Vorwort

Die Gesellschaft für Informatik e.V. (GI) vergibt gemeinsam mit der Schweizer Informatik Gesellschaft (SI) und der Österreichischen Computergesellschaft (OCG) jährlich einen Preis für eine hervorragende Dissertation im Bereich der Informatik, die einen wesentlichen Beitrag zur Weiterentwicklung der Informatik und deren Anwendungsgebieten oder zum Verständnis der Wechselwirkungen zwischen Informatik und Gesellschaft leistet. Jede deutsche, österreichische und schweizer Universität und Hochschule mit Promotionsrecht kann eine ihrer Dissertationen des vorangegangenen Jahres für diesen Preis nominieren. Für das Jahr 2021 wurden 31 Dissertationen eingereicht. Deren Autoren und Autorinnen haben damit bereits eine besondere Würdigung ihrer Hochschule erfahren.

Im Rahmen des Auswahlprozesses wird ein Kolloquium veranstaltet. Dies konnte nach Abklingen der Pandemie erstmalig wieder in Präsenz im Leibniz-Zentrum für Informatik Schloss Dagstuhl durchgeführt werden. Fast alle Nominierten haben daran teilgenommen und ihre innovativen Methoden und Ergebnisse präsentiert. Sehr hoch waren wiederum die Breite der Themen und das Niveau der Vorträge. An jeden Vortrag schlossen sich spezifische Nachfragen und eine kurze Diskussion an. Zusätzlich hatten die Nominierten ausgiebig Möglichkeiten, sich untereinander informell auszutauschen.

Wie jedesmal, aber in diesem Jahr besonders fiel es der Jury schwer, *eine* Dissertation für den GI-Preis auszuwählen. Mit einer Kurzfassung der nominierten Dissertationen in diesem Band sollen alle eine angemessene Würdigung erfahren und einer breiten Öffentlichkeit vorgestellt werden. Damit ist auch ein Beitrag zum Wissenstransfer innerhalb der Informatik und von den Universitäten und Hochschulen in die Bereiche Technik, Wirtschaft und Gesellschaft beabsichtigt.

Besonders beeindruckend waren dies Jahr eine Reihe von Dissertationen in den Gebieten Algorithmik, IT-Sicherheit und Maschinelles Lernen. In diesen Bereichen hat das Auswahlgremium jeweils eine Arbeit als besonders preiswürdig identifiziert. Eine weitere Differenzierung zwischen diesen drei Spitzenreitern war auch nach intensiver Diskussion diesmal nicht möglich. Jede Auswahl hätte die anderen zu Unrecht benachteiligt. Somit teilen sich 2021 drei Dissertationen den ersten Platz. Ausgezeichnet werden:

Dr. Markus Hecher für seine Dissertation „Werkzeuge und Methoden zum Lösen von Problemen mittels Baumweite“. Herr Hecher hat mit neuen Beweismethoden die Komplexität von Erfüllbarkeitsproblemen bei beschränkter Baumweite der Formeln präzise analysiert sowie Algorithmen entwickelt und implementiert, die sich trotz der hohen worst-case Komplexität als praxistauglich auch für große Probleminstanzen erweisen und anderen Verfahren auf Standard-Benchmarks überlegen sind.

Dr. Moritz Lipp für seine Dissertation „Die Ausnutzung von Optimierungen in Mikroarchitekturen durch Software“. Herr Lipp war maßgeblich beteiligt an der Aufdeckung gravierender Sicherheitslücken in Betriebssystemen moderner Mikroprozessoren und an Maßnahmen, diese zu schließen. Seine Ergebnisse werden prägenden Einfluss auf die zukünftige Entwicklung von System-Architekturen haben.

Dr. Alejandro Molina Ramirez für seine Dissertation „Tiefe Netzwerke, die wissen, wenn sie etwas nicht wissen“. Herr Molina Ramirez hebt mit seinen Ergebnissen den Ein-

satz von probabilistischen und kausalen Modellen im Maschinellen Lernen auf eine neue Ebene. Mit Hilfe seiner mathematisch stringenten Analyse können erstmals auch quantitative Aussagen über die Zuverlässigkeit von Vorhersagen gewonnen werden.

Mit diesen Preisverleihungen werden gewürdigt:

- eine herausragende algorithmische Arbeit, die ein zentrales komplexitätstheoretisches Problem exakt löst und daraus sehr effiziente praxistaugliche Entscheidungsverfahren ableitet
- ein Meilenstein in der IT-Sicherheitsforschung, der einen riesigen Impact auf Industrie und Gesellschaft generiert hat
- ein Quantensprung im Maschinellen Lernen, der vielen Anwendern eine ganz neue Erkenntnistiefe ermöglicht

Ein großer Dank gilt dem Auswahlgremium für sein Engagement bei dieser zeitaufwändigen und anspruchsvollen Aufgabe, insbesondere seinem Vorsitzenden Prof. Dr. Steffen Hölldobler. Leider war es Herrn Hölldobler nicht möglich, das diesjährige Verfahren zum Abschluss zu bringen. Daher habe ich diese Aufgabe nun übernommen.

Des weiteren möchte ich mich bedanken bei Frau Sylvia Wünsch für die Organisation der online-Vorträge, Frau Dr. Lena Reinfelder und Herrn Stefan Sobernig für die Zusammenstellung des Bandes und der Geschäftsstelle der Gesellschaft für Informatik e.V. für die technische Unterstützung des Auswahlverfahrens und schließlich bei dem gesamten Team von Schloss Dagstuhl für das perfekte Ambiente während des Kolloquiums.

Rüdiger Reischuk
Lübeck im September 2022

Abbildung 1: Teilnehmer*innen des Kolloquiums

Abbildung 2: Mitglieder des Auswahlausschusses

Kandidat*innen für den GI-Dissertationspreis 2021

Dr. Da Silva, Carina	Westfälische Wilhelms-Universität Münster
Dr. rer. nat. Dubslaff, Clemens	Technische Universität Dresden
Dr.-Ing. Eberhardt, Jacob	Technische Universität Berlin
Dr. Gabor, Thomas	Ludwig-Maximilians-Universität München
Dr. Gnad, Daniel	Universität des Saarlandes
Dr.-Ing. Gossen, Frederik Jakob	Technische Universität Dortmund
Dr. Hassan, Muhammad	Universität Bremen
Dr. Hayat, Samira	Universität Klagenfurt
Dr. techn. Hecher, Markus	Technische Universität Wien
Dr.-Ing. Klare, Heiko	Karlsruher Institut für Technologie
Dr. Kolberg, Jascha	Hochschule Darmstadt
Dr. Kottke, Daniel	Universität Kassel
Dr. Kraus, Matthias	Universität Konstanz
Dr.-Ing. Krüger, Jacob	Ruhr-Universität Bochum
Dr. Lauscher, Anne	Universität Mannheim
Dr. techn. Lipp, Moritz	Technische Universität Graz
Dr. Lorenz, Jan-Hendrik	Universität Ulm
Dr. Luo, Linghui	Universität Paderborn
Dr. Mair, Sebastian	Leuphana Universität Lüneburg
Dr. Meggendorfer, Tobias	Technische Universität München
Dr. Molina Ramirez, Alejandro	Technische Universität Darmstadt
Dr. Müller-Budack, Eric	Leibniz Universität Hannover
Dr.-Ing. Roschke, Christian	Hochschule Mittweida
Dr. Schaller, David	Universität Leipzig
Dr. Suleri, Sarah	RWTH Aachen
Dr. Svozil, Alexander	Universität Wien
Dr. Szabo, Tamas	Johannes-Gutenberg-Universität Mainz
Dr. Thyagarajan, Sri Aravinda K.	Friedrich-Alexander-Universität Erlangen-Nbg.
Dr. Uzunova, Hristina	Universität zu Lübeck
Dr. Weninger, Markus	Johannis Kepler Universität Linz
Dr. Zambon, Daniele	Universita della Svizzera italiana

Mitglieder des Auswahlausschusses für den GI-Dissertationspreis 2021

Prof. Dr. Steffen Hölldobler (Vorsitzender bis Juni 2022)	Technische Universität Dresden
Prof. Dr. Rüdiger Reischuk (Vorsitzender ab Juli 2022)	Universität zu Lübeck
Prof. Dr. Sven Apel	Universität des Saarlandes
Prof. Dr. Abraham Bernstein	Universität Zürich
Prof. Dr.-Ing. Felix Freiling	Universität Erlangen-Nürnberg
Prof. Dr. Hans-Peter Lenhof	Universität des Saarlandes
Prof. Dr. Gustaf Neumann	Wirtschaftsuniversität Wien
Prof. Dr. Kay Uwe Römer	TU Graz
Prof. Dr. Björn Scheuermann	Humbold-Universität zu Berlin
Prof. Dr. Nicole Schweikardt	Humbold-Universität zu Berlin
Prof. Dr. Klaus Wehrle	RWTH Aachen

Inhaltsverzeichnis

Statistisches Model Checking und zeitlich begrenzte Erreichbarkeitsanalyse für hybride Petri-Netze mit mehreren stochastischen Variablen[1]

Carina da Silva[2]

Abstract: Sicherheitskritische Systeme stellen einen wichtigen Teil des heutigen Lebens dar. Modellierung und formale Verifikation bieten Ansätze zur Analyse solcher Systeme im Hinblick auf Systemeigenschaften wie beispielsweise Zuverlässigkeit. In meiner Dissertation [Pi21] wird eine Unterklasse stochastischer hybrider Systeme betrachtet, die diskrete, kontinuierliche und stochastische Variablen kombiniert. Es werden neuartige Ansätze für die Evaluation von *hybriden Petri-Netzen mit allgemeinen Transitionen* (HPnGs) vorgestellt. Diese umfassen statistisches Model Checking für Modelle mit linearen und nichtlinearen kontinuierlichen Verläufen sowie die (zeitlich begrenzte) Erreichbarkeitsanalyse für nichtdeterministische Modelle. Darüber hinaus stellt die Dissertation einen Ansatz für eine Transformation von HPnGs in eine Unterklasse der stochastischen hybriden Automaten vor, die die Anwendung bestehender, für hybride Automaten entwickelter Methoden auf stochastische hybride Modelle ermöglicht. Der resultierende Fehler der vorgestellten Ansätze kann dabei genau charakterisiert werden.

1 Einleitung

Sicherheitskritische Systeme sind Systeme, deren Fehlverhalten oder Ausfall verheerende Konsequenzen haben können. Dazu zählen unter anderem kritische Infrastrukturen (wie zum Beipsiel Strom- und Gastnetzwerke), nukleare Systeme, die medizinische Versorgung und das Transportwesen. Aufgrund ihrer Bedeutung und wachsenden Komplexität wächst der Bedarf an Methoden zur Evaluation sicherheitskritischer Systeme. Die Untersuchung solcher Systeme in der Praxis erweist sich jedoch oft als schwierig oder gar unmöglich. Hier bietet Modellierung einen beliebten alternativen Ansatz, bei dem nicht in das reale System eingegriffen wird. Mithilfe von Methoden des *Model Checkings* [BK08] können bestehende Modelle automatisiert auf Systemeigenschaften wie Zuverlässigkeit, Verfügbarkeit und Sicherheit überprüft werden. Da sich Modellklassen in ihrer Syntax und Semantik unterscheiden, gibt es eine Vielzahl von Model Checking Ansätzen.

Die Analyse *stochastischer hybrider Systeme* ist aufgrund der Ausdrucksstärke dieser mächtigen Modellklasse eine anspruchsvolle Aufgabe. Bestehende Ansätze skalieren häufig

[1] Englischer Titel der Dissertation: "Statistical Model Checking and Time-Bounded Reachability Analysis for Hybrid Petri Nets with Multiple Stochastic Variables'

[2] Westfälische Wilhelms-Universität Münster, Institut für Informatik, Einsteinstraße 62, 48149 Münster, Deutschland carina.dasilva@uni-muenster.de

nicht angemessen und basieren auf Abstraktionen, die zu einem Kompromiss zwischen Genauigkeit und Effizienz führen. Meine Dissertation [Pi21] stellt verschiedene Methoden zur Analyse sogenannter *hybrider Petri-Netze mit allgemeinen Transitionen* (HPnGs) [GR16] vor, die von statistischem Model Checking bis zur formalen Verifikation reichen. Der resultierende Fehler der Ansätze kann dabei charakterisiert werden.

HPnGs wurden bereits erfolgreich für die Analyse einer Wasseraufbereitungsanlage [GRH13] und in den Bereichen Smart Home [Gh15] und Elektromobilität [HR16] eingesetzt. Allerdings bergen bestehende Analyseansätze für HPnGs die Gefahr einer „Explosion“ des Zustandsraumes, da jede induzierte Zufallsvariable den Zustandsraum eines HPnGs um eine Dimension erweitert. Dies schränkt die Skalierbarkeit der Ansätze in der Praxis stark ein. Als Alternative zu Analyseverfahren stellt die meine Dissertation statistisches Model Checking für HPnGs vor und erweitert dieses Konzept für HPnGs mit nichtlinearem kontinuierlichen Verhalten, das durch Systeme gewöhnlicher Differentialgleichungen ausgedrückt wird. Des Weiteren stellt die Arbeit einen Ansatz für die zeitlich begrenzte Erreichbarkeitsanalyse von HPnGs vor, bei dem exakte Mengen erreichbarer Zustände bestimmt werden. Zur Berechnung der Wahrscheinlichkeit, definierte Zielzustände zu erreichen, wird dabei auf numerische Integration zurückgegriffen. Für den Umgang mit diskretem Nichtdeterminismus werden nichtprophetische und prophetische Strategien betrachtet.

Darüber hinaus wird eine Transformation von HPnGs in eine Unterklasse von stochastischen hybriden Automaten vorgestellt. Diese ermöglicht die Verwendung bestehender, für die Verifikation hybrider Automaten entwickelter Methoden für stochastische hybride Systeme. Diese Methoden werden in der Dissertation zu einer *Flowpipe*-basierten Erreichbarkeitsanalyse für die aus der Transformation resultierende Modellklasse erweitert. Dieses Analyseverfahren ist ebenfalls bis auf den Fehler der numerischen Integration exakt.

2 Stochastische hybride Systeme

Wir unterscheiden Systeme anhand der Art der Variablen, die zur Beschreibung der Eigenschaften eines Systems vonnöten sind. Ein diskretes System wird ausschließlich durch diskrete Variablen beschrieben, die Werte aus einer endlichen oder abzählbar unendlichen Menge annehmen. Dies kann zum Beispiel der binäre Zustand eines Geräts sein oder die Häufigkeit, wie oft ein bestimmtes Ereignis eintritt. In einem kontinuierlichen System nehmen die Variablen Werte aus einer unendlichen Menge an. Beispiele für kontinuierliche Variablen sind die Flüssigkeitsmenge in einem Gefäß oder physikalische Größen wie Temperatur oder Druck. Systeme, die sowohl diskrete als auch kontinuierliche Variablen kombinieren, bezeichnen wir als *hybrid*.

Ein System mit probabilistischem oder stochastischem Verhalten wird durch zufällige Ereignisse, wie zum Beispiel Systemausfälle, bestimmt. Das Auftreten dieser Ereignisse kann durch diskrete oder kontinuierliche Zufallsvariablen modelliert werden, denen (diskrete

oder kontinuierliche) Wahrscheinlichkeitsverteilungen zugeordnet sind. Wir verstehen dabei *stochastische* Variablen als von der Zeit abhängige Zufallsvariablen.

Es ist möglich, dass ein System in einem Modell nicht vollständig spezifiziert ist – sei es absichtlich oder aufgrund fehlender Informationen. Solche Modelle bezeichnen wir als nichtdeterministisch. Wir unterscheiden dabei zwischen diskretem und kontinuierlichem Nichtdeterminismus in Abhängigkeit davon, ob sich die Unterspezifikation auf diskrete oder kontinuierliche Variablen bezieht.

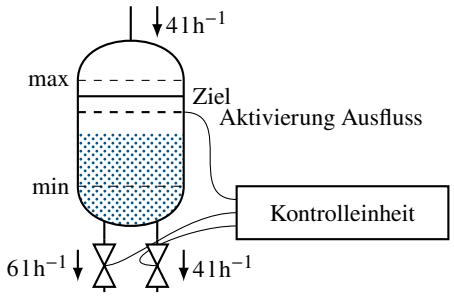


Abb. 1: Tank mit konstantem Einfluss und zwei kontrollierten Ausflussventilen [PSR21, Abb. 4].

Beispiel Abb. 1 zeigt die Skizze eines Tanks, der durch eine Pumpe dauerhaft gefüllt und durch zwei ein- und ausschaltbare Ventile entleert wird [PSR21, Abb. 4]. Die Flüssigkeit im Tank wird durch eine kontinuierliche Variable und die Zustände der Ventile durch zwei diskrete Variablen beschrieben. Die Kontrolleinheit des Tanks kann wählen, welches Ventil aktiviert werden soll, was zu einer diskreten nichtdeterministischen Entscheidung führt. Nach Deaktivierung sind beide Ventile für eine zufällige Zeitspanne blockiert, wobei die Dauer jeder Blockierungsphase durch eine eigene kontinuierliche Zufallsvariable beschrieben wird. Somit ist dieses System ein Beispiel für ein hybrides System mit stochastischem Verhalten und diskretem Nichtdeterminismus. Ein solcher Tank könnte Teil eines sicherheitskritischen Systems sein, in dem ein Über- oder Unterlauf der Flüssigkeit im Tank katastrophale Schäden verursacht.

Hybride Petri-Netz mit allgemeinen Transitionen Unsere Definition für HPnGs folgt Gribaudo und Remke [GR16]. Ein *hybrides Petri-Netz mit allgemeinen Transitionen* ist als ein Tupel $(\mathcal{P}, \mathcal{T}, \mathcal{A}, \mathbf{M}_0, \Phi)$ definiert. Die Menge $\mathcal{P} = \mathcal{P}^{disc} \cup \mathcal{P}^{cont}$ ist eine endliche Menge von diskreten und kontinuierlichen *Plätzen*. Die diskreten Plätze eines HPnGs enthalten eine natürliche Anzahl von *Marken*. Die kontinuierlichen Plätze enthalten dagegen eine *Flüssigkeit*, deren Flüssigkeitsstand durch einen reellen Wert ausgedrückt wird. Jede Markierung $\mathbf{M}_i$ des HPnGs setzt sich dabei aus einer Markenbelegung $\mathbf{m}_i$ und Flüssigkeitsständen $\mathbf{x}_i$ zusammen. $\mathbf{M}_0$ bezeichnet die initiale Markierung.

Die Menge $\mathcal{T} = \mathcal{T}^{imm} \cup \mathcal{T}^{det} \cup \mathcal{T}^{gen} \cup \mathcal{T}^{cont}$ ist eine endliche Menge von *Transitionen*. Wir unterscheiden zwischen unmittelbaren, deterministischen, allgemeinen und kontinuierlichen Transitionen. Plätze und Transitionen werden durch Kanten der Menge $\mathcal{A} = \mathcal{A}^{disc} \cup \mathcal{A}^{cont} \cup \mathcal{A}^{test} \cup \mathcal{A}^{inh}$ verbunden. Diskrete Kanten aus $\mathcal{A}^{disc}$ verbinden diskrete Plätze mit nichtkontinuierlichen Transitionen; kontinuierliche Kanten aus $\mathcal{A}^{cont}$ verbinden kontinuierliche Plätze und kontinuierliche Transitionen. Durch das *Feuern* einer Transition kann sich die aktuelle Markierung der mit ihr verbundenen Plätze ändern. Eine Test- oder Hemmkante aus $\mathcal{A}^{test}$ bzw. $\mathcal{A}^{inh}$ ermöglicht dabei das Kontrollieren einer Transition in Abhängigkeit der Markierung des mit ihr verbundenen Platzes. Alle nichtkontinuierlichen Transitionen ändern die Markenbelegung diskreter Plätze zu ihrem Feuerungszeitpunkt. Allgemeine Transitionen aus $\mathcal{T}^{gen}$ haben dabei zufällige Feuerungszeiten, die durch Zufallsvariablen beschrieben werden. Diese folgen absolut kontinuierlichen Wahrscheinlichkeitsverteilungen. Kontinuierliche Transitionen hingegen verändern die Flüssigkeitsstände kontinuierlich über die Zeit mit stückweise-konstanten Flußraten.

Die Details zur Semantik von HPnGs werden in der Dissertation erläutert. Die Parameterfunktionen aus dem Tupel Φ dienen dabei der Festlegung von Kapazitätsgrenzen, Feuerungszeiten, Flussraten für die kontinuierlichen Transitionen sowie Verteilungsregeln für Konflikte unter Transitionen und werden in dieser Kurzfassung nicht weiter ausgeführt.

3 Beiträge der Dissertation zum Stand der Wissenschaft

Die Beiträge der Dissertation zum Stand der Wissenschaft lassen sich in drei Teilgebiete aufteilen, die im Folgenden erläutert werden: statistisches Model Checking für HPnGs (Abschnitt 3.1), zeitlich begrenzte Erreichbarkeitsanalyse für HPnGs (Abschnitt 3.2) und der Übergang zu stochastischen hybriden Automaten (Abschnitt 3.3).

3.1 Statistisches Model Checking für HPnGs

Meine Dissertation präsentiert Algorithmen für die ereignisbasierte Simulation von HPnGs, basierend auf dem in meiner Masterarbeit [Pi16] entwickelten Ansatz. Dabei wird in jedem Simulationslauf ein Zufallswert für jede Feuerung einer allgemeinen Transition generiert. Zur statistischen Auswertung von Eigenschaften in HPnGs dient sogenanntes *statistisches Model Checking*. Dabei lässt sich der maximale resultierende statistische Fehler spezifizieren, der wiederum die Anzahl der erforderlichen Simulationsläufe bestimmt.

Wir möchten die Wahrscheinlichkeit, dass zu einem festen Zeitpunkt eine bestimmte definierte Eigenschaft in einem HPnG erfüllt ist, abschätzen. Da Simulation mit einer endlichen Anzahl an Simulationsläufen nie ein exaktes Ergebnis liefert, wird der Anteil der möglichen Verläufe, die die Eigenschaft erfüllen, über ein Konfidenzintervall abgeschätzt, sodass für ein Irrtumsniveau $\alpha \in [0, 1]$ die tatsächliche Wahrscheinlichkeit in $(100 \cdot (1 - \alpha))$

Prozent der Fälle von dem Konfidenzintervall überdeckt wird. Für diese Methode betrachtet die Dissertation vier Ansätze: das Standard-, das Wald-, das Clopper-Pearson- und das Score-Konfidenzintervall.

Eine Alternative zur Berechnung von Konfidenzintervallen bieten Hypothesentests. Diese entscheiden, ob die Wahrscheinlichkeit, dass die betrachtete Eigenschaft erfüllt ist, größer oder kleiner als ein definierter Schwellenwert ist. Die Dissertation betrachtet den sequentiellen Likelihood-Quotienten-Test, den Gauss Single Sampling Plan, den Gauss Confidence Interval Test, den Chow-Robbins Test und den Azuma-Test. Diese Tests bringen jeweils verschiedene Vorteile und Risiken mit sich, die in der Arbeit diskutiert werden.

Die in der Dissertation vorgestellten Algorithmen wurden in dem Simulator HYPEG[3] implementiert, mit dessen Hilfe die verschiedenen Arten von Konfidenzintervallen und Hypothesentests in der Dissertation im Rahmen einer Fallstudie über einen Ladevorgang für Elektrofahrzeuge miteinander verglichen werden.

Simulation mit nichtlinearen kontinuierlichen Verläufen Die Dissertation erweitert den Modellformalismus der HPnGs um nichtlineare kontinuierliche Verläufe, die durch Systeme gewöhnlicher Differentialgleichungen beschrieben werden können. Dazu wird die Menge der Transitionen um sogenannte dynamische kontinuierliche Transitionen erweitert, deren aktuelle Flußrate von den aktuellen Werten der kontinuierlichen Variablen, also der Flüssigkeitstände, abhängen kann.

Dementsprechend werden in der Dissertation die Methoden des statistischen Model Checkings für nichtlineare Modelle erweitert, wobei jedoch für die Bestimmung des Zeitpunktes des jeweils nächsten Ereignisses in der Simulation eine Approximation erforderlich ist. Um das nichtlineare Verhalten abzuschätzen, wird der kontinuierliche Teil des HPnGs in ein sogenanntes *quantisiertes Zustandssystem zweiter Ordnung* (QSS2) nach Kofman [Ko02] überführt. In diesem System wird jede kontinuierliche Variable durch einen Quantisierer erster Ordnung approximiert. Dabei ist sichergestellt, dass die lokale Differenz zwischen dem tatsächlichen Wert der kontinuierlichen Variablen und dem Wert des Quantisierers einen vordefinierten Schwellenwert nicht überschreitet. Sobald die lokale Differenz diesen Schwellenwert erreicht, wird der Wert des Quantisierers neu berechnet. Für stabile lineare zeitinvariante Systeme ist dabei der Fehler im zugehörigen quantisierten Zustandssystem zweiter Ordnung begrenzt und abhängig vom gewählten Schwellenwert. Abb. 2 skizziert einen möglichen Verlauf einer Variablen $x_i(t)$ und deren Quantisierer $q_i(t)$ über die Zeit t, unter Berücksichtigung des Schwellenwerts Δq_i.

Für das statistische Model Checking von HPnGs mit nichtlinearen Verläufen wird der Simulationsansatz in der Dissertation um die Neuberechnung der Quantisierer als zusätzliche Ereignisart erweitert. Für den Fall, dass beim Überprüfen einer Eigenschaft die Approximation zu ungenau ist, werden zusätzliche Ereignisse, also frühere Neuberechnungen

[3] https://zivgitlab.uni-muenster.de/ag-sks/tools/hypeg

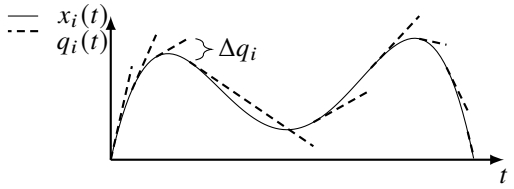


Abb. 2: Eingangs- und Ausgangstrajektorien in einem Quantisierer erster Ordnung, eigene Darstellung nach [Ko02, p.79, Figure 4].

des Quantisierers, ergänzt. Dieser Ansatz wird in der Dissertation in einer Fallstudie über das kinetische Batteriemodell [MM93] für nichtlineares Lade- und Entladeverhalten von Batterien validiert.

3.2 Zeitlich begrenzte Erreichbarkeitsanalyse

Meine Dissertation stellt einen Ansatz für die zeitlich begrenzte Erreichbarkeitsanalyse von HPnGs vor, der auf der symbolischen Zustandsraumdarstellung des sogenannten *Parametric Location Tree* (PLT) [Hü21] basiert. Diese Darstellung beschreibt das stochastische Verhalten eines HPnGs symbolisch in Abhängigkeit der Zufallsvariablen. Dabei werden in einer Baumstruktur Zustände mit identischen Werten der diskreten Variablen in Knoten zusammengefasst. Durch die Begrenzung der Zeit ist der resultierende Baum endlich und vollständig berechenbar.

Die zeitlich begrenzte Erreichbarkeitsanalyse bestimmt exakte Mengen von erreichbaren Zuständen basierend auf dem Parametric Location Tree und verwendet Monte-Carlo-Methoden für mehrdimensionale Integration zur Berechnung der Wahrscheinlichkeit, definierte Zielzustände innerhalb einer Zeitspanne $[0, t_{\max}], t_{\max} \in \mathbb{R}_{\geq 0}$ zu erreichen. Jeder Knoten Λ_k des Parametric Location Tree enthält für jede Zufallsvariable $s_i, i \in \{1, \ldots, n\}$ einen Wertebereich $[l_i, u_i]$, sodass der Knoten Λ_k genau dann erreicht wird, wenn alle n Zufallsvariablen einen Wert innerhalb des jeweiligen Bereichs annehmen. Die Wahrscheinlichkeit $p(\Lambda_k, t_{\max})$ einen Zielknoten Λ_k innerhalb der Zeitspanne $[0, t_{\max}]$ zu erreichen, wird dabei via mehrdimensionaler Integration wie folgt berechnet:

$$p(\Lambda_k, t_{\max}) = \int_{l_1}^{u_1} \int_{l_2}^{u_2} \ldots \int_{l_n}^{u_n} \prod_{i=1}^{n} g_i\,(s_i)\;\, ds_n \ldots\; ds_2\; ds_1, \tag{1}$$

wobei für $i \in \{1, \ldots, n\}$, g_i die zur Zufallsvariablen s_i gehörige Wahrscheinlichkeitsdichtefunktion ist. Für den letzten Schritt der numerischen Integration wird der statistische Fehler in der Dissertation abgeschätzt.

Strategien Wenn in einem HPnG zwei oder mehr deterministische oder unmittelbare Transitionen zum selben Zeitpunkt feuern sollen, entsteht diskreter Nichtdeterminismus.

Während bestehende Analyseverfahren für HPnGs einen solchen inhärenten diskreten Nichtdeterminismus durch Prioritäten und Gewichte probabilistisch auflösen, bleibt er in der zeitlich begrenzten Erreichbarkeitsanalyse erhalten. Die Dissertation untersucht stattdessen verschiedene Arten von Strategien (engl. *scheduler*), um nichtdeterministische Entscheidungen zu treffen. Meine Dissertation stellt Algorithmen zur Bestimmung maximaler und minimaler Wahrscheinlichkeiten für zeitlich begrenzte Erreichbarkeit vor, wobei über die Klassen der sogenannten nichtprophetischen und prophetischen Strategien optimiert wird.

Aufgrund der Verwendung des Parametric Location Tree können nichtprophetische Strategien dabei vom diskreten Teil aller vorherigen Zuständen, der sogenannten diskreten Vergangenheit, abhängen. Dabei betrachten wir für einen Zustand Γ_k eine Menge $C(\Gamma_k) \subseteq \mathcal{T}^{det} \cup \mathcal{T}^{imm}$ von deterministischen oder unmittelbaren Transitionen, die zum selben Zeitpunkt feuern sollen. Eine solche von der diskreten Vergangenheit abhängige, *nichtprophetische Strategie* ist definiert als eine messbare Funktion $\mathfrak{s}$, die jeder diskreten Vergangenheit, die zu einem Konflikt von Transitionen der Menge $C(\Gamma_k)$ im Zustand Γ_k führt, eine diskrete Wahrscheinlichkeitsverteilung über die Transitionen in $C(\Gamma_k)$ zuweist.

Da ein Pfad durch den Parametric Location Tree die diskrete Vergangenheit kodiert und für die nichtprophetische Strategie keine weiteren Informationen verfügbar sind, trifft eine solche Strategie immer dieselbe Entscheidung in einem Knoten des Baumes. Daher kann die maximale bzw. minimale Wahrscheinlichkeit, einen Zielzustand zu erreichen, bestimmt werden, indem der Baum rekursiv iteriert wird: Für jeden Konflikt können die Erreichbarkeitswahrscheinlichkeiten der entsprechenden Teilbäume verglichen und das jeweilige Maximum bzw. Minimum von unten nach oben gereicht werden, sodass die optimale Wahrscheinlichkeit letztendlich an die Wurzel gereicht wird.

Prophetische Strategien berücksichtigen hingegen zusätzlich zukünftige Feuerungszeiten der allgemeinen Transitionen. Sie „kennen“ also die Werte der Zufallsvariablen. Prophetische Strategien können (aufgrund ihres Wissens über die Zufallsvariablen und der symbolischen Darstellung des PLT) von der gesamten Vergangenheit abhängen. Sei für ein gegebenes HPnG, n die (maximale) Anzahl an Zufallsvariablen, die in diesem HPnG bis zu einer Zeitgrenze induziert werden. Eine von der Vergangenheit abhängige, *prophetische Strategie* für ein HPnG ist somit eine messbare Funktion $\mathfrak{s}$, die jedem Paar aus einer Vergangenheit, die zu einem Konflikt von Transitionen der Menge $C(\Gamma_k)$ im Zustand Γ_k führt, und einer Zuordnung von n positiven reelen Werten zu den Zufallsvariablen eine diskrete Wahrscheinlichkeitsverteilung über die Transitionen in $C(\Gamma_k)$ zuweist.

Da in einem HPnG per Definition keine Informationen über zukünftige Feuerungszeiten der allgemeinen Transitionen verfügbar sind, ist für die Berücksichtigung prophetischer Strategien eine Modellanpassung nötig. Diese ermöglicht die Vorberechnung der Feuerungszeiten und das Speichern dieser Zeiten mithilfe zusätzlicher kontinuierlicher Variablen. Für das angepasste Modell können dann optimale Wahrscheinlichkeiten, die prophetischen Strategien entsprechen, berechnet werden. Dafür wird ebenfalls der Parametric Location Tree iteriert

und ein resultierendes Optimierungsproblem gelöst. Die Details zur Modellanpassung und zum Optimierungsproblem sind in der Dissertation dargelegt.

Die Algorithmen für optimale nichtprophetische Strategien wurden in dem Tool hpnmg[4] implementiert. Für den prophetischen Fall wurde ein Machbarkeitsnachweis erbracht, der jedoch die Verfügbarkeit eines effizienten Solvers für das zugrunde liegende Optimierungsproblem erfordert. In der Dissertation wird die Realisierbarkeit beider Ansätze in einer Fallstudie über eine Entscheidung zwischen einem Fahrzeug mit Elektroantrieb und einem Fahrzeug mit Verbrennungsmotor demonstriert.

3.3 Übergang zu stochastischen hybriden Automaten

Meine Dissertation stellt eine Transformation von HPnGs in eine Unterklasse stochastische hybrider Automaten vor, die die Anwendung bestehender analytischer Methoden ermöglicht. Diese Transformation basiert auf der Definition einer Semantik für ein gegebenes HPnG, die durch einen *singulären Automaten mit Zufallsuhren und zwingenden Sprüngen* ausgedrückt wird.

Wir folgen der Definition hybrider Automaten von Alur et al. [Al95]. Ein singulärer Automat ist ein hybrider Automat, in dem die erste Ableitung einer jeden kontinuierlichen Variablen konstant ist. Dabei wird in der Dissertation durch die Restriktion auf zwingende Sprünge kontinuierlicher Determinismus ausgeschlossen. Diskrete Zustandsänderungen erfolgen also nur zu festen Zeitpunkten. Die Erweiterung von singulären Automaten um Zufallsuhren ermöglicht die Modellierung von stochastischem Verhalten. Die Zeit bis zum Ablauf einer Zufallsuhr wird durch eine absolut kontinuierliche Wahrscheinlichkeitsverteilung beschrieben. Eine formale Definition dieser Modellklasse ist in der Dissertation gegeben.

Ein Algorithmus für die Überführung eines HPnGs in einen solchen Automaten wird in der Dissertation vorgestellt und wurde ebenfalls in das hpnmg-Tool implementiert. Die Machbarkeit wird in der Dissertation für einen Batterieladeprozess demonstriert.

Flowpipe-basierte Erreichbarkeitsanalyse Für die Unterklasse stochastischer hybrider Automaten, die wir durch die Transformation erhalten, stellt meine Dissertation eine neue Methode zur Berechnung von Wahrscheinlichkeiten für zeitlich begrenzten Erreichbarkeit vor. Der Ansatz basiert auf der Konstruktion einer sogenannten *Flowpipe* [Fr05], welche ein bewährtes Analyseverfahren für hybride Automaten ist. Dieser Ansatz liefert die Menge aller erreichbaren Zustände. Darauf aufbauend stellt meine Dissertation Algorithmen für die Berechnung optimaler Wahrscheinlichkeiten unter Berücksichtigung vergangenheitsabhängiger nichtprophetischer und prophetischer Strategien für singuläre Automaten mit Zufallsuhren vor. Durch eine geometrische Repräsentation der Zustandsmengen als konvexe Polytope kann dabei auf geometrische Operationen zurückgegriffen werden, die es

[4] https://zivgitlab.uni-muenster.de/ag-sks/tools/hpnmg

ermöglichen, nichtprophetische und prophetische Wahrscheinlichkeiten mit vergleichbarem Aufwand zu berechnen. Die Menge der erreichbaren Zustände ist dabei exakt, wohingegen zur Berechnung von Wahrscheinlichkeiten erneut numerische Integration dient, sodass eine Abschätzung des statistischen Fehlers möglich ist. Die Komplexität der Algorithmen hängt dabei stark von den geometrischen Operationen ab.

Meine Dissertation validiert die nichtprophetischen Ergebnisse in einer Fallstudie über den Tank mit zwei Ventilen (siehe Abb. 1) mit Ergebnissen des Analyseansatzes für HPnGs. Für den prophetischen Fall wird die Machbarkeit anhand desselben Modells gezeigt, für das bisher kein anderes Tool in der Lage war Ergebnisse für höhere Dimensionen zu berechnen. Die Ergebnisse zeigen, dass die Algorithmen sowohl für die nichtprophetischen als auch für die prophetischen Strategien gleich effizient sind, wobei ihre Skalierbarkeit von der Effizienz der zugrunde liegenden Darstellung der Zustandsmengen abhängt.

4 Schlussbemerkung

Die in der Dissertation vorgestellten Methoden bieten neue Ansätze für die Evaluation der mächtigen Klasse der stochastischen hybriden Systeme, indem die Analyse von hybriden Petri-Netzen mit stochastischen Variablen mit Methoden der Erreichbarkeitsanalyse für hybride Automaten vereint wird. Die Verfahren erlauben dabei eine angemessene Abschätzung der induzierten Fehler. Die Ergebnisse der Fallstudien in der Arbeit demonstrieren, dass die Implementierungen auch für höhere Dimensionen präzise Ergebnisse liefern, wobei die Performanz von der Verfügbarkeit effizienter Solver für die Optimierung und numerische Integration sowie von skalierbaren Zustandsraumdarstellungen für hybride Automaten abhängt. Durch die Entwicklung der vorgestellten Methoden leistet die Dissertation einen Beitrag zur Analyse sicherheitskritischer Systeme.

Literatur

[Al95] Alur, R.; Courcoubetis, C. A.; Halbwachs, N.; Henzinger, T. A.; Ho, P.-H.; Nicollin, X.; Olivero, A.; Sifakis, J.; Yovine, S.: The Algorithmic Analysis of Hybrid Systems. Theoretical Computer Science 138/, S. 3–34, 1995.

[BK08] Baier, C.; Katoen, J.-P.: Principles of Model Checking. MIT Press, Cambridge, MA, USA, 2008.

[Fr05] Frehse, G.: PHAVer: Algorithmic Verification of Hybrid Systems Past HyTech. In: Proc. 8th Int. Workshop on Hybrid Systems: Computation and Control. Bd. 3414. LNCS, Springer Berlin Heidelberg, S. 258–273, 2005.

[Gh15] Ghasemieh, H.; Haverkort, B. R.; Jongerden, M. R.; Remke, A.: Energy Resilience Modelling for Smart Houses. In: Proc. 45th IEEE/IFIP Int. Conf. on Dependable Systems and Networks. IEEE, S. 275–286, 2015.

[GR16] Gribaudo, M.; Remke, A.: Hybrid Petri Nets with General One-Shot Transitions. Performance Evaluation 105/, S. 22–50, 2016.

[GRH13] Ghasemieh, H.; Remke, A.; Haverkort, B. R.: Survivability Evaluation of Fluid Critical Infrastructures Using Hybrid Petri Nets. In: Proc. 19th IEEE Pacific Rim Int. Symposium on Dependable Computing. IEEE, S. 152–161, 2013.

[HR16] Hüls, J.; Remke, A.: Coordinated Charging Strategies for Plug-in Electric Vehicles to Ensure a Robust Charging Process. In: Proc. 10th EAI Int. Conf. on Performance Evaluation Methodologies and Tools. ICST, S. 19–22, 2016.

[Hü21] Hüls, J.; Pilch, C.; Schinke, P.; Niehaus, H.; Delicaris, J.; Remke, A.: State-Space Construction of Hybrid Petri Nets with Multiple Stochastic Firings. ACM Transactions on Modeling and Computer Simulation 31/3, S. 1–37, 2021.

[Ko02] Kofman, E.: A Second-Order Approximation for DEVS Simulation of Continuous Systems. SIMULATION 78/2, S. 76–89, 2002.

[MM93] Manwell, J. F.; McGowan, J. G.: Lead Acid Battery Storage Model for Hybrid Energy Systems. Solar Energy 50/5, S. 399–405, 1993.

[Pi16] Pilch, C.: Development of an Event-Based Simulator for Model Checking Hybrid Petri Nets with Random Variables, Masterarbeit, Westfälische Wilhelms-Universität Münster, 2016.

[Pi21] Pilch, C.: Statistical Model Checking and Time-Bounded Reachability Analysis for Hybrid Petri Nets with Multiple Stochastic Variables, Dissertation, Westfälische Wilhelms-Universität Münster, 2021.

[PSR21] Pilch, C.; Schupp, S.; Remke, A.: Optimizing Reachability Probabilities for a Restricted Class of Stochastic Hybrid Automata via Flowpipe-Construction. In: Proc. 18th Int. Conf. on Quantitative Evaluation of Systems. Bd. 12846. LNCS, Springer Cham, S. 435–456, 2021.

Carina da Silva geb. Pilch wurde am 28. Juni 1990 in Rheine, Deutschland, geboren. Sie schloss 2012 ihr duales Studium der Wirtschaftsinformatik (Bachelor) an der Hochschule Osnabrück als Beste ihres Studiengangs ab. Von 2014 bis 2016 studierte sie Informatik im Master an der Westfälischen Wilhelms-Universität Münster (WWU). Für ihre Masterarbeit über diskrete Ereignissimulation für hybride Petri-Netze mit Zufallsvariablen erhielt sie 2017 den *Preis des Fakultätentags Informatik für eine herausragende Masterarbeit*. Im Anschluss an ihr Studium erhielt sie das *Ada Lovelace Promotionsstudium 2016* des Fachbereichs Mathematik und Informatik der WWU. Ihre Promotion unter Betreuung von Prof. Dr. Anne Remke über statistisches Model Checking und zeitlich begrenzte Erreichbarkeitsanalyse für hybride Petri-Netze mit Zufallsvariablen schloss sie im Oktober 2021 mit Auszeichnung (*summe cum laude*) ab. Seit Februar 2022 ist sie als akademische Rätin am Institut für Informatik der WWU tätig.

Quantitative Analyse von konfigurierbaren und rekonfigurierbaren Systemen[1]

Clemens Dubslaff[2]

Abstract: Die Fülle an Konfigurationsoptionen und der daraus resultierende Reichtum an Systemvarianten stellen Entwickler von modernen Computersystemen vor großen Herausforderungen. Weitere Systemanforderungen an Adaptivität, Rekonfigurierbarkeit und an quantitative Aspekte wie Zuverlässigkeit, Energieverbrauch oder Latenz kommen erschwerend hinzu. Formale Analysen sind daher unabdingbar, um die Auswirkungen von Konfigurationsoptionen und deren Interaktionen einzuschätzen und fehlerfreie Systeme zu garantieren. Die vorgestellte Dissertation führt ein kompositionelles Modellierungs- und Analyseframework ein, welches alle genannten Herausforderungen adressiert und effektive Lösungen bietet, formale quantitative Analysen auch für bisher unmöglich große konfigurierbare Systeme durchzuführen. An real existierenden Systemen wird dessen Anwendbarkeit demonstriert und mit neuen Methoden zu kausalen Erklärungen von Analyseresultaten ergänzt.

1 Einleitung

Fast jedes Softwaresystem ist heutzutage konfigurierbar. Angefangen auf der Compilerebene, in der z.B. durch `#ifdef`-Optionen spezielle Optimierungen für die verwendete Hardware aktiviert werden können, bis hin zur Produktebene, bei der Kunden verschiedene Varianten der gleichen Software mit unterschiedlichsten Funktionalitäten zur Auswahl stehen. Die Zahl der möglichen Systemvarianten ist hierbei häufig exponentiell in der Anzahl der Konfigurationsoptionen. Dies stellt insbesondere Software- und Systementwickler vor große Herausforderungen, denn sie müssen für die Vorhersage, Erkennung, Beschreibung und das Beseitigen von Fehlern und ungewolltem Verhalten eine Vielzahl von Kombinationen von Konfigurationen betrachten. Das Verändern von Konfigurationen zur Laufzeit des Systems, sogenannte *Rekonfigurationen*, erhöhen die Komplexität nochmals. Diese sind z.B. in modernen Softwaresystemen durch Updates, Einflüsse von Nutzern und deren Einstellungen, sowie durch Selbstadaptivität und sich verändernden Umgebungen in natürlicher Weise gegeben. Neben der Konfigurierbarkeit und Rekonfigurierbarkeit von Software stellt deren Einbettung in cyber-physische Systeme (CPS) eine weitere Quelle von Komplexität dar. Das Verhalten von CPS hängt von *quantitativen Aspekten* wie dem Energieverbrauch, der Bandbreite oder Fehlerwahrscheinlichkeiten ab, welche allesamt in neuen Technologien wie 5G-Netzwerken, dem taktile Internet, sowie dem autonomen Fahren eine zentrale Rolle spielen. Um diesen Quellen der Komplexität zu begegnen und fehlerfreie Systeme

[1] Englischer Titel der Dissertation: "Quantitative Analysis of Configurable and Reconfigurable Systems"

[2] Technische Universität Dresden, Institut für Theoretische Informatik, Nöthnitzer Straße 46, 01187 Dresden, Deutschland; email: clemens.dubslaff@tu-dresden.de

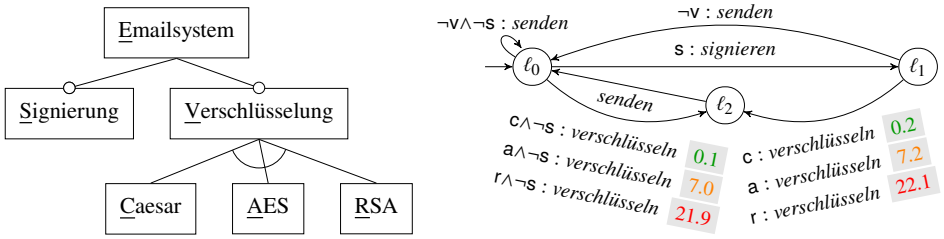


Abb. 1: Featurediagramm (links) und featured Transitionssystem mit Kostenannotationen (rechts)

zu entwickeln, werden verschiedenste Analysemethoden eingesetzt. Simulative Ansätze und Testmethoden können zeigen, dass Fehler oder Systemeigenschaften beim Ausführen des Systems möglich sind. Dahingehend zeigen Verifikationstechniken die Abwesenheit von Fehlern und geben Garantien an Systemeigenschaften, sind aber meist nur für Kernkomponenten und kleine Systeme praktikabel. Eine weit verbreitete Verifikationsmethode ist *Model Checking* [z.B. BK08], bei der das System und dessen Anforderungen formal spezifiziert und erschöpfend analysiert wird. *Probabilistisches Model Checking (PMC)* unterstützt zudem die Betrachtung von Wahrscheinlichkeiten und quantitativen Größen und wird erfolgreich zur quantitativen Analyse von einer Vielzahl von Systemen angewendet. Symbolische Methoden basierend auf *binären Entscheidungsdiagrammen (BDDs)* können Model Checking auch für reale Computersysteme ermöglichen [BK08].

Um konfigurierbare Systeme zu spezifizieren und analysieren, stellen *Features* ein weit verbreitetes Konzept dar. Sie kapseln optionale oder inkrementelle Funktionalitäten und wurden bisher in erster Linie für den Entwurf und Entwicklung von Software-Produktlinien (SPLs) verwendet [Ap13]. Systemvarianten ergeben sich durch die Auswahl von Features und korrespondieren somit zu Konfigurationen als Mengen von Features. Üblicher Weise werden feature-orientierte Systeme mittels eines zweistufigen Ansatzes spezifiziert:

(1) Basierend auf einer Domänenanalyse werden Features isoliert und deren Beziehungen in einem *Variabilitätsmodell* ausgedrückt.
(2) Das *Verhalten von Features* wird spezifiziert, z.B. in einer geeigneten Programmiersprache welche operationelle Abhängigkeiten zwischen Features beschreiben kann.

Ein gängiges Variabilitätsmodell stellen Featurediagramme dar [Ka90]. In solchen Diagrammen werden die konfigurationsrelevante Abhängigkeiten von Features in einer hierarchischen Baumstruktur modelliert, bei denen Kindfeatures die Auswahl von Elternfeatures implizieren. In Abb. 1 ist ein Featurediagramm für ein konfigurierbares Emailsystem dargestellt. Dieses System enthält optionale Signierungs- und Verschlüsslungsfeatures (modelliert durch einen Kreis oberhalb des Features), sowie Verschlüsslungsarten Caesar, AES und RSA, von denen bei ausgewähltem Verschlüsslungsfeature genau eine Art ausgewählt werden muss (modelliert durch die verbundene Verzweigung zwischen den Verschlüsslungsarten). Durch das Featurediagramm wird eine Systemfamilie mit acht Systemvarianten beschrieben: ohne Verschlüsslung oder mit einer der drei Verschlüsslungsarten, je mit Signierungsfeature oder

ohne. Ein Beispiel ist die Systemvariante ohne Signierung und mit AES Verschlüsslung, welche durch die Konfiguration $\{\mathsf{e}, \mathsf{v}, \mathsf{a}\}$ als Menge von Features formalisiert wird.
Für die Spezifikation von Verhalten konfigurierbarer Systeme unterscheidet man zwischen *annotativen* und *kompositionellen* Ansätzen [KA08]. Beide Ansätze haben komplementäre Vor- und Nachteile bezüglich Granularität, Erweiterbarkeit und Analyse. Mittels annotativer Methoden werden Verhalten mit aussagenlogischen Ausdrücken über Features versehen. Diese Verhalten sind nur in jenen Konfigurationen aktiv, welche den annotierten Featureausdruck erfüllen. Beispiele hierfür sind *featured Transitionssysteme (FTS)* [Cl13], bei denen Zustandsübergänge mit Featureausdrücken annotiert werden, und `#ifdef`-Optionen in C-Programmen. In Abb. 1 ist auf der rechten Seite ein FTS für das konfigurierbare Emailsystem dargestellt. Startend im Zustand ℓ_0 kann eine Email nur dann direkt versendet werden, wenn weder das Signierungs- noch das Verschlüsselungsfeature ausgewählt wurde (formalisiert durch den Featureausdruck $\neg\mathsf{v}\wedge\neg\mathsf{s}$). Ansonsten muss eine Email zunächst signiert oder verschlüsselt werden (Transitionen von ℓ_0 nach ℓ_1 oder ℓ_2). Annotative Ansätze bieten feingranulare Spezifikationsmöglichkeiten und erlauben eine *familienbasierte Analyse*, in der alle Systemvarianten in einem Analyseschritt betrachtet werden und symbolische Methoden Gemeinsamkeiten zwischen Systemvarianten ausnutzen können. Kompositionelle Modellierungsansätze für konfigurierbare Systeme beschreiben das Verhalten jedes Features isoliert als *Featuremodul*. Das Gesamtverhalten einer Systemvariante entsteht durch die Komposition der Featuremodule von ausgewählten Features mittels eines Kompositionsoperators. Diese Methode wird hauptsächlich in der Entwicklung von SPLs angewendet [Ap13], wobei als Kompositionsoperator die *Superimposition* eingesetzt wird. Superimposition beschreibt wie ein Featuremodul das Verhalten eines Basissystems verändert. Im FTS Beispiel von Abb. 1 würde durch das Signierungsfeature z.B. das direkte Versenden einer Mail in ℓ_0 durch eine Transition nach ℓ_1 ersetzt werden. Kompositionelle Ansätze haben Vorteile in der Trennung von Zuständigkeiten, Modularisierung und der daraus resultierenden Wart- und Erweiterbarkeit des konfigurierbaren Systems.

Aufgrund der komplementären Vor- und Nachteile annotativer und kompositioneller Methoden, stellten Kästner und Apel die Frage, ob ein *hybrider* Ansatz möglich ist, der die Vorteile beider Methoden vereint [KA08]. Die Dissertation [Du21] liefert mehrere fundamentale Beiträge zur Spezifikation und (quantitativen) Analyse von konfigurierbaren Systemen, indem u.a. ein hybriden Ansatz vorgestellt wird und dessen Vorteile formal bewiesen werden. Beiträge sind unter anderem:

(1) Ein *annotativ kompositionelles Framework* zur Spezifikation und Analyse von konfigurierbaren Systemen, u.a. für quantitative Erweiterungen von FTS.
(2) *Kompositionelle Familienmodelle*, welche eine familienbasierte Analyse auch für kompositionelle Ansätze ermöglichen.
(3) Modellierung und Analyse *rekonfigurierbarer Systeme* mit Hilfe von Familienmodellen.
(4) *Reduktion von Analyseproblemen* für konfigurierbare Systeme auf Standardanalyseprobleme und deren algorithmische Lösung.
(5) Neue Methoden zur *Reduktion von Modellgrößen*, insbesondere für Familienmodelle.

(6) Neuartige Auswertung von Analyseresultaten in konfigurierbaren Systemen durch *kausale Analyse* auf der Abstraktionsebene der Features.
(7) Anwendung und Evaluation auf reale (Hardware-)Systeme.

2 Konfigurierbare probabilistische Systeme

Kern der Dissertation [Du21] bildet die Formalisierung eines hybriden Ansatzes zur Spezifikation und Analyse konfigurierbarer Systeme mit quantitativen Aspekten.

Definition 1 *Ein* annotatives kompositionelles System (ACS) *ist ein Tupel* $\mathsf{Acs} = (F, \mathcal{V}, \mathfrak{M}, \phi, \prec, \circ)$ *über eine Menge von Features F, einem Variabilitätsmodell $\mathcal{V}$, einer Menge von annotierten Featuremodulen $\mathfrak{M}$, einer Zuordnung von Featuremodulen zu Features über eine Funktion $\phi\colon F \to \mathfrak{M}$, sowie eine totale Kompositionsordnung über Features $\prec$ und eine Kompositionsoperation $\circ$.*

Die Dissertation betrachtet hauptsächlich Featuremodule als annotierte Programme in einer feature-orientierten und probabilistischen Variante von Dijkstras *Guarded Commands*. Das präsentierte Framework ist jedoch generisch und die erbrachten Ergebnisse lassen sich auf eine Vielzahl von Formalismen übertragen. Für diese Zusammenfassung reicht es, sich Featuremodule als quantitative Varianten von FTS vorzustellen [DBK15; DKB14]. Ein Beispiel ist das FTS in Abb. 1, bei dem Transitionen mit zu den erwartenden Verschlüsselungszeiten annotiert sind. Als Kompositionsoperationen fokussieren wir uns auf feature-orientierte Varianten der parallelen Komposition $\|$ und Superimposition $\bullet$ [Du19a].

Das Verhalten einer Systemvariante ergibt sich für ACS mittels Komposition und Projektion. Werden Features $X \subseteq F$ ausgewählt, so ist die korrespondierende Systemvariante $\phi(X)\restriction_X$, wobei $\phi(X) = \phi(x_1) \circ \phi(x_2) \circ \ldots \circ \phi(x_k)$ für $X = \{x_1, x_2, \ldots, x_k\}$ mit $x_1 \prec x_2 \prec \ldots \prec x_k$ die Komposition von Featuremodulen für die Features in X gemäß der Featureordnung entspricht und die Projektion $\phi(X)\restriction_X$ all jene Verhalten in $\phi(X)$ entfernt, deren Featureausdrücke nicht in X erfüllt sind. Demnach ist der ACS Ansatz im Kern kompositionell, erlaubt aber auch feingranulare Bedingungen durch annotative Elemente.

2.1 Kompositionelle Familienmodelle

Waren Familienmodelle bisher nur für annotative Verfahren bekannt, werden diese auch für kompositionelle Ansätze in der Dissertation eingeführt. Zugrunde liegt eine erstaunlich einfache Beobachtung: eine Spezifikation, welche die Verhalten aller Featuremodule beinhalten soll, benötigt die Komposition aller Featuremodule für den gesamten Konfigurationsraum.

Definition 2 *Ein ACS* $\mathsf{Acs} = (F, \mathcal{V}, \mathfrak{M}, \phi, \prec, \circ)$ *ist ein* kompositionelles Familienmodell *wenn $\phi(F)\restriction_X = \phi(X)\restriction_X$ für alle validen Konfigurationen $X \subseteq F$.*

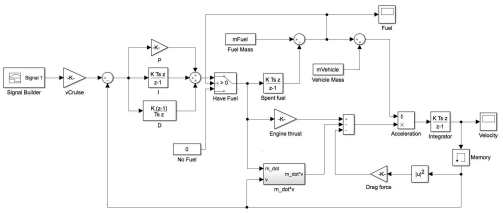
Während in annotativen Ansätzen generell von einem Familienmodell ausgegangen wird, ist nicht jedes ACS a priori ein kompositionelles Familienmodell. Grund sind mögliche Interaktionen zwischen Features, die in den validen Systemvarianten nicht auftauchen, jedoch aber bei einer Komposition aller Featuremodule. Im Emailbeispiel von Abb. 1 könnte es z.B. geschehen, dass unvorhergesehen mehrere Verschlüsselungsmethoden auch nach Projektion ausführbar sind, weil Featuremodule unabhängig voneinander entwickelt wurden.

Theorem 1 *Für jedes* $\circ$*-ACS mit* $\circ \in \{\|, \bullet\}$ *kann in polynomieller Zeit ein kompositionelles Familienmodell mit gleichen Systemvarianten konstruiert werden.*

Familienmodelle sind insbesondere für die familienbasierte Analyse unter Verwendung von symbolischen Methoden von Vorteil [Th14]. Symbolische Methoden, z.B. mittels BDDs, nutzen Gemeinsamkeiten zwischen Systemvarianten aus und können die exponentiell vielen Systemvarianten komprimiert darstellen und analysieren. Neben dieser bekannten Rolle von Familienmodellen stellt die Dissertation einen neuen Vorteil vor. Rekonfigurationen können in Familienmodellen elegant modelliert werden, da das Modell alle Verhalten und somit auch die Verhalten nach den Rekonfigurationen beinhaltet. Die Grundidee hierbei ist, die Rekonfigurationen im Variabilitätsmodell mit einzubeziehen: Zustände in diesem Variabilitätsmodell stehen für valide Konfigurationen und Transitionen beschreiben Rekonfigurationen. Ein solches Modell deckt den statischen Fall ebenfalls ab, indem valide Konfigurationen initial und keine Rekonfigurationen modelliert sind. Für ein ACS Acs löst ein spezieller (paralleler) Operator $\bowtie$ in Verbindung mit dem Familienmodell $\phi(F)$ Featureausdrücke auf und führt zur *Semantik* $\phi(F) \bowtie \mathcal{V}$ von Acs. Diese Semantik ist ein klassischen Programm, welches die Systemkonfigurationen mit in den Zuständen kodiert. Damit reduziert sich die funktionale und quantitative Analyse von (re)konfigurierbaren Systemen auf Standardmethoden, welche direkt auf $\phi(F) \bowtie \mathcal{V}$ angewendet werden können. Dies komplementiert Methoden der Literatur, welche Featureannotationen speziell in die Analysemethoden mit einbeziehen [Cl13; Th14] und somit für jede neuartige Analysemethode wiederholt auf konfigurierbare Systeme angepasst werden müssen. Zudem waren bisher keine familienbasierte Analysemethoden für rekonfigurierbare Systeme bekannt und existierende Modellierungsmethoden benötigten die Spezifikation spezieller Regeln für die Aktivierung und Deaktivierung von Features.

2.2 Zwischen Kompositionswelten

Während Parallelkompositionen $\|$ hauptsächlich in Beschreibungssprachen zur formalen Analyse verwendet werden, stellt die Superimposition $\bullet$ die Standardkomposition für feature-orientierte Softwareentwicklung dar. In der Dissertation werden beide Kompositionsoperatoren im Zusammenhang mit ACS betrachtet. Hierbei stellt sich die natürliche Frage, ob ACS mit unterschiedlichen Kompositionsoperatoren ineinander überführbar sind, d.h., in ein ACS mit gleicher Semantik $\phi(F) \bowtie \mathcal{V}$. Solche Transformationen hätten den Vorteil, feature-orientierte Programme der Softwareentwicklung in Programme zur



ϑ_R	MFunc	Sum1, Sum4	Memory	MOne, P, vCruise, SpentFuel, I	D	Prod	DragForce, EThrust	Integrator	Sum5	Sum3	Sum2	Acceleration
0.019	0.63	1.00	0.63	1.00	1.00	1.00	0.63	1.00	1.00	1.00	1.00	1.00
0.020	0.70	0.80	0.77	0.82	0.94	1.00	0.70	0.82	1.00	0.80	0.80	0.80	1.00
0.022	0.49	0.55	0.55	0.61	0.75	0.95	0.49	0.61	1.00	0.55	0.55	0.55	1.00
0.025	0.35	0.38	0.36	0.40	0.46	0.65	0.35	0.40	1.00	0.38	0.38	0.38	1.00
0.029	0.28	0.29	0.28	0.29	0.31	0.33	0.28	0.29	0.91	0.29	0.29	0.29	1.00
0.034	0.28	0.29	0.30	0.30	0.32	0.38	0.28	0.30	0.57	0.29	0.30	0.30	0.97
0.040	0.26	0.27	0.27	0.27	0.28	0.31	0.26	0.27	0.37	0.27	0.34	0.34	0.86
0.047	0.24	0.25	0.25	0.25	0.25	0.26	0.25	0.25	0.30	0.25	0.36	0.36	0.78
0.055	0.24	0.24	0.24	0.24	0.25	0.26	0.24	0.25	0.27	0.26	0.37	0.39	0.68
0.064	0.23	0.23	0.23	0.23	0.24	0.24	0.24	0.25	0.27	0.27	0.35	0.41	0.61

Abb. 2: SIMULINK Geschwindigkeitsregler (links) und Fehlerschuldgrade der Komponenten (rechts)

formalen Analyse umzuwandeln und umgekehrt, Prototypen und Gerüste von konfigurierbarer Software mittels formaler Sprache zu entwickeln und zu verifizieren, bevor diese in feature-orientierte Software umgewandelt und verfeinert werden.

Theorem 2 *Für jedes $\|$-ACS gibt es ein exponentiell großes $\bullet$-ACS mit der gleichen Semantik. Umgekehrt gibt es $\bullet$-ACS für die es kein $\|$-ACS mit der gleichen Semantik gibt. Für jedes $\bullet$-ACS gibt es jedoch ein exponentiell großes $\|$-ACS mit den gleichem Verhalten.*

3 Zuverlässigkeitsanalyse in Redundanzsystemen

Wir haben in Abschnitt 2 gesehen, wie die familienbasierte Analyse von (re)konfigurierbaren Systemen auf Standardmethoden zurück geführt werden kann. Dies ermöglicht die Anwendung moderner Analysemethoden, w.z.B. Kosten-Nutzen-Analysen mittels PMC [BDK14] oder PMC für wissensintensive Systeme [DKT19]. Hierbei sei auch auf PROFEAT [Ch18], die Implementierung des in Abschnitt 2 vorgestellten Frameworks verwiesen, welche für $\|$-ACS die nötigen Transformationen automatisiert durchführt. Um die Vorteile des Frameworks zu demonstrieren, wurden in der Dissertation [Du21] verschiedenste Fallstudien u.a. an realen Systemen durchgeführt. Hierbei war auch der Fokus auf die familienbasierte quantitative Analyse von *Hardwaresystemen*. Während für Softwaresysteme die Vorteile einer familienbasierten Analyse bekannt waren, wurde Hardware bisher nur marginal untersucht. Neben der Parametersynthese und Analyse in rekonfigurierbaren Netzwerksystemen [BD18; DBK15; DKB14] wurden insbesondere SIMULINK-*Redundanzsysteme* eingeführt und untersucht [Du19b; Du20a; Du20b]. In solchen Systemen treten Komponenten redundant auf, um die Fehlertoleranz des Gesamtsystems zu erhöhen. Am Bekanntesten ist der *TMR-Mechanismus*, welcher die Ergebnisse von drei redundanten Komponenten mittels einer Majoritätsfunktion vergleicht. Aufgefasst als konfigurierbares System steht die Einführung von TMR für ein Feature. Die geringste Fehlerwahrscheinlichkeit kann natürlich erreicht werden, indem alle TMR Features aktiv sind. Jedoch führt die Einführung von TMR zu größeren Ausführungszeiten, höheren Energieverbrauch, sowie größeren Chips in Hardwareimplementierungen. Ein zentrales Ziel bei der Entwicklung von Redundanzsystemen ist es somit, unter gegebenen Kostenanforderungen jene Konfiguration zu ermitteln, welche

die höchste Zuverlässigkeit garantiert. In der Dissertation [Du21] wurde insbesondere eine Methode entwickelt, SimuLink Modelle in das feature-orientierte Framework von Abschnitt 2 zu übersetzen. Hierbei wurde auch ein Flugzeuggeschwindigkeitsregler, wie in Abb. 2 links dargestellt, und weitere Redundanzmechanismen untersucht. Fortschrittliches symbolisches PMC ermöglichte die Berechnung von Fehlerwahrscheinlichkeiten für mehr als 10^{10} Konfigurationen innerhalb von weniger als zwei Stunden auf einem handelsüblichen Rechner. Mit mehr als $2,5 \cdot 10^{13}$ Zuständen stellt dieses Modell auch eines der größten erfolgreich verifizierter realer Systeme dar und wurde in die MARS Benchmark Datenbank aufgenommen [Du20a].
Für die Konstruktion und Analyse des Reglermodells waren neuartige Reduktionstechniken nötig, welche in der Dissertation [Du21] eingeführt werden. *Iterative Variablenumordnung* [Du20a] ordnet Variablen in BDDs so um, dass sie Gemeinsamkeiten im Familienmodell besser komprimieren. Hierbei werden sukzessive Features zur Familie hinzugefügt, kombiniert mit klassischen Umordnungstechniken [Kl18]. Inspiriert von Techniken der Compileroptimierung verringern *Zurücksetzungs- und Registerallokationsreduktion* [Du20b] die Größe des Zustandsraums des Familienmodells.

4 Kausalität in konfigurierbaren Systemen

Die potenziell exponentielle Anzahl von Systemvarianten und der daraus folgenden großen Anzahl von Analyseergebnissen oder Fehlerberichten erfordert besondere Techniken für deren sinnvolle und praktikable Beschreibung. In der Dissertation [Du21] wurden hierzu grundlegende Konzepte und Methoden mittels *kausalem Schließen auf der Ebene von Features* präsentiert [Ba21; Du22]. Der neue Begriff der *Featuregründe* beschreibt hierbei jene Teilkonfigurationen von Features, die ein Grund für *Effekte* in Form von emergentem Systemverhalten sind. Dieser stützt sich auf die prominente kontrafaktische Definition von Kausalität von Halpern und Pearl [HP01]. Da Features meist auch eine intuitive Bedeutung den Systemfunktionalitäten zuordnen, können Erklärungen für Analyseergebnisse auf der Ebene von Features wichtige Erkenntnisse zur Systementwicklung geben [Ap13]. Effekte werden hierbei als Mengen von validen Konfigurationen beschrieben, wobei sowohl funktionale als auch quantitative Eigenschaften Effekte erzielen können. Im vorherigen Emailbeispiel könnte der Effekt “lange Verschlüsselungszeit” z.B. mit einer Menge $\mathsf{Eff}_{>5} \subseteq 2^F$ von validen Konfigurationen beschrieben werden, bei denen die erwartete Verschlüsselung länger als 5 Zeiteinheiten benötigt. Gemäß Abb. 1 ist $\mathsf{Eff}_{>5} = \big\{\{\mathsf{e},\mathsf{v},\mathsf{a}\}, \{\mathsf{e},\mathsf{v},\mathsf{r}\}, \{\mathsf{e},\mathsf{v},\mathsf{s},\mathsf{a}\}, \{\mathsf{e},\mathsf{v},\mathsf{s},\mathsf{r}\}\big\}$, da nur bei Caesar Verschlüsslung nicht mehr als 5 Zeiteinheiten benötigt werden.

Definition 3 *Ein* Featuregrund *für valide Konfigurationen* $\mathsf{Eff} \subseteq 2^F$ *ist ein Paar* $G = (G_0, G_1)$ *von inaktiven Features* $G_0 \subseteq F$ *und aktiven Features* $G_1 \subseteq F$*, sodass*

(1) $\varnothing \neq [\![G]\!] \subseteq \mathsf{Eff}$ *für die Menge* $[\![G]\!]$ *aller validen Konfigurationen bei denen alle Features aus* G_0 *nicht ausgewählt und alle Features aus* G_1 *ausgewählt sind, und*

(2) G minimal ist, d.h., Entfernen eines Features aus G_0 *oder* G_1 *führt zu* G' *mit* $[\![G']\!] \not\subseteq \mathsf{Eff}$.

Die erste Bedingung (1) formalisiert, dass alle vom Grund induzierten validen Konfigurationen auch den Effekt zeigen, während (2) die kontrafaktische Schlussfolgerung enthält: sollte eine vom Featuregrund gestellte Bedingung an ein Feature wegfallen, so ist es möglich, dass der Effekt nicht mehr eintritt. Im Emailbeispiel erhält man so drei Featuregründe für $\mathsf{Eff}_{>5}$: $(\varnothing, \{\mathsf{a}\})$, $(\varnothing, \{\mathsf{r}\})$, und $(\{\mathsf{c}\}, \{\mathsf{v}\})$. Die ersten beiden Gründe sind intuitiv und offensichtlich, denn AES oder RSA Verschlüsselung sind verantwortlich für lange Verschlüsselungszeiten. Der dritte Grund, der explizit eine Verschlüsslung fordert, aber die Caesar-Methode ausschließt, zeigt, dass manche Gründe nicht sofort offensichtlich sind aber sogar mehrere valide Konfigurationen abdecken können. In der Dissertation wird bewiesen, dass Featuregründe mit hinreichenden Primimplikanten der Effekt- und nicht-validen Konfigurationsmenge übereinstimmen, woraus sich direkt ein Algorithmus zu deren Berechnung ergibt.

Theorem 3 *Die Menge aller Featuregründe ist in polynomieller Zeit in der Anzahl der Systemkonfigurationen berechenbar.*

Da die Anzahl der Systemkonfigurationen jedoch exponentiell in der Anzahl der Features ist und es auch exponentiell viele Featuregründe geben kann, sind weitere Verfahren nützlich, die Featuregründe zur Erklärung von Effekten weiter verarbeiten. In der Dissertation werden neue Methoden zur Reduktion von aussagenlogischen Formeln für die Featuregrundmenge, als auch Effektüberdeckungen, *Verantwortlichkeiten* und *Schuldgrade* [CH04], sowie Featureinteraktionen vorgestellt. Zudem wurde mittels mehrerer Experimente aus dem Bereich der Analyse von konfigurierbaren Systemen gezeigt, dass Featuregründe beim Aufspüren von Fehlern und zum Erklären von Effekten nützlich sind.
Exemplarisch betrachten wir hier Schuldgrade, welche auch zur Erklärung der Verlässlichkeit des Reglers von Abb. 2 verwendet wurden. Die Verantwortlichkeit eines Features ist bezüglich einer gegebenen Effektkonfiguration, dem Kontext, definiert. Sie steht für den Anteil an Features, deren Belegung zusätzlich zu diesem Feature geändert werden müssen, um ein kontrafaktisches Beispiel zu erzeugen, d.h., den Effekt nicht mehr zu zeigen. Ein Wert von 1 steht somit für volle Verantwortlichkeit des Features, während ein Wert von 0 für keine Verantwortlichkeit steht. Schuldgrade nehmen eine globale Sicht auf Effektkonfigurationen ein und geben die zu erwartende Verantwortlichkeit eines Features wider. Die Tabelle in Abb. 2 listet die Schuldgrade für die Komponenten des in Abschnitt 3 analysierten Reglers für verschiedenste Zuverlässigkeitsschranken ϑ_R auf. So ist die Einführung von TMR für “Integrator” und “Acceleration”-Komponenten jeweils voll verantwortlich für den Effekt einer Fehlerwahrscheinlichkeit von weniger als $2,5\%$, was durch Standardmethoden der quantitative Analyse nicht direkt ableitbar ist. Solche Erkenntnisse können verwendet werden, um Faustregeln für die Konfiguration von Systemen abzuleiten. Im Reglerbeispiel sollten Systementwickler für geringe Fehlerwahrscheinlichkeiten TMR-Konfigurationen mit “Integrator” und “Acceleration”-Komponenten verwenden.

Literatur

[Ap13] Apel, S.; Batory, D. S.; Kästner, C.; Saake, G.: Feature-Oriented Software Product Lines - Concepts and Implementation. Springer, 2013.

[Ba21] Baier, C.; Dubslaff, C.; Funke, F.; Jantsch, S.; Majumdar, R.; Piribauer, J.; Ziemek, R.: From Verification to Causality-Based Explications. In: Proc. of the 48th International Colloquium on Automata, Languages, and Programming (ICALP). Bd. LIPIcs:198, Leibniz-Zentrum für Informatik, S. 1–20, 2021.

[BD18] Baier, C.; Dubslaff, C.: From Verification to Synthesis under Cost-Utility Constraints. ACM SIGLOG News 5/4, S. 26–46, 2018.

[BDK14] Baier, C.; Dubslaff, C.; Klüppelholz, S.: Trade-off Analysis Meets Probabilistic Model Checking. In: Proc. of the 23rd Conf. on Computer Sci. Logic and the 29th Symp. on Logic In Computer Sci. (CSL-LICS). ACM, S. 1–10, 2014.

[BK08] Baier, C.; Katoen, J.-P.: Principles of Model Checking. The MIT Press, 2008.

[CH04] Chockler, H.; Halpern, J. Y.: Responsibility and Blame: A Structural-Model Approach. Artificial Intelligence Research 22/, S. 93–115, 2004.

[Ch18] Chrszon, P.; Dubslaff, C.; Klüppelholz, S.; Baier, C.: ProFeat: feature-oriented engineering for family-based probabilistic model checking. Formal Aspects of Computing 30/, S. 45–75, 2018.

[Cl13] Classen, A.; Cordy, M.; Schobbens, P.-Y.; Heymans, P.; Legay, A.; Raskin, J.-F.: Featured Transition Systems: Foundations for Verifying Variability-Intensive Systems and Their Application to LTL Model Checking. Transactions on Software Engineering 39/, S. 1069–1089, 2013.

[DBK15] Dubslaff, C.; Baier, C.; Klüppelholz, S.: Probabilistic Model Checking for Feature-Oriented Systems. Transactions on Aspect-Oriented Software Development LNCS:8989/, S. 180–220, 2015.

[DKB14] Dubslaff, C.; Klüppelholz, S.; Baier, C.: Probabilistic Model Checking for Energy Analysis in Software Product Lines. In: Proc. of the 13th Conf. on Modularity (MODULARITY). ACM, S. 169–180, 2014.

[DKT19] Dubslaff, C.; Koopmann, P.; Turhan, A.-Y.: Ontology-Mediated Probabilistic Model Checking. In: Proceedings of the 15th Conference on integrated Formal Methods (iFM). Bd. LNCS:11918, Springer, S. 194–211, 2019.

[Du19a] Dubslaff, C.: Compositional Feature-Oriented Systems. In: Proceedings of the 17th Conference on Software Engineering and Formal Methods (SEFM). Bd. LNCS:12226, Springer, S. 162–180, 2019.

[Du19b] Dubslaff, C.; Ding, K.; Morozov, A.; Baier, C.; Janschek, K.: Breaking the Limits of Redundancy Systems Analysis. In: Proc. of the 29th European Safety and Reliability Conf. (ESREL). S. 2317–2325, 2019.

[Du20a] Dubslaff, C.; Morozov, A.; Baier, C.; Janschek, K.: Iterative Variable Reordering: Taming Huge System Families. In: Proc. of the 4th Workshop on Models for Formal Analysis of Real Systems (MARS). Bd. EPTCS:316, S. 121–133, 2020.

[Du20b] Dubslaff, C.; Morozov, A.; Baier, C.; Janschek, K.: Reduction Methods on Error-Propagation Graphs for Quantitative Systems Reliability Analysis. In: Proc. of the 30th European Safety and Reliability Conf. and 15th Probabilistic Safety Assessment and Management Conf. (ESREL-PSAM). 2020.

[Du21] Dubslaff, C.: Quantitative Analysis of Configurable and Reconfigurable Systems, Diss., TU Dresden, Institute for Theoretical Computer Science, 2021.

[Du22] Dubslaff, C.; Weis, K.; Baier, C.; Apel, S.: Causality in Configurable Software Systems. In: Proc. of the 44th Intern. Conf. on Software Engineering (ICSE). ACM, Pittsburgh, Pennsylvania, S. 325–337, 2022.

[HP01] Halpern, J. Y.; Pearl, J.: Causes and Explanations: A Structural-Model Approach - Part I: Causes. In: Proc. of the 17th Conf. in Uncertainty in Artificial Intelligence (UAI). Morgan Kaufmann, S. 194–202, 2001.

[KA08] Kästner, C.; Apel, S.: Integrating compositional and annotative approaches for product line engineering. In: Proc. GPCE Workshop on Modularization, Comp. and Generative Techniques for Product Line Engineering. S. 35–40, 2008.

[Ka90] Kang, K. C.; Cohen, S. G.; Hess, J. A.; Novak, W. E.; Peterson, A. S.: Feature-Oriented Domain Analysis (FODA) Feasibility Study, Techn. Ber., Carnegie-Mellon University Software Engineering Institute, 1990.

[Kl18] Klein, J.; Baier, C.; Chrszon, P.; Daum, M.; Dubslaff, C.; Klüppelholz, S.; Märcker, S.; Müller, D.: Advances in probabilistic model checking with PRISM: variable reordering, quantiles and weak deterministic Büchi automata. Software Tools for Technology Transfer 20/, S. 179–194, 2018.

[Th14] Thüm, T.; Apel, S.; Kästner, C.; Schaefer, I.; Saake, G.: A Classification and Survey of Analysis Strategies for Software Product Lines. Computing Surveys 47/, 6:1–6:45, 2014.

Clemens Dubslaff studierte Mathematik und Informatik an der TU Dresden, bevor er u.a. an der Neuen Universität Lissabon (Portugal) einen Master in “Computational Logic” ablegte. Nach einem Auslandsaufenthalt am NICTA in Sydney (Australien) kehrte er für seine Promotion an die TU Dresden zurück. Er ist nun an der TU Eindhoven (Niederlande) im Bereich der formalen Systemanalyse tätig. Seine wissenschaftliche Arbeit ist breit gefächert und umfasst sowohl rein theoretische, als auch interdisziplinäre und softwaretechnologische Beiträge.

Skalierbare und vertraulichkeitswahrende Off-Chain Berechnungen[1]

Jacob Eberhardt[2]

Abstract: Blockchains erlauben sich gegenseitig mistrauenden Parteien gemeinsame Transaktionen auszuführen und deren Historie unveränderlich zu speichern. Aufgrund ihres technischen Aufbaus leiden Blockchains jedoch unter niedrigem Durchsatz, fehlender Skalierbarkeit und schwachen Datenschutzgarantien. Diese Arbeit adressiert diese Probleme durch das neuartige Konzept des Off-Chainings: Daten und Berechnungen werden von einer Blockchain auf externe Resourcen ausgelagert - jedoch ohne dabei Schlüsseleigenschaften der Blockchain zu kompromittieren. Insbesondere verifizierbare Off-Chain Berechnungen stellen ein mächtiges Werkzeug zur Erhöhung des Durchsatzes und der Gewährleistung von Vertraulichkeit dar. Allerdings fehlen geignete Realisierungsansätze. Unsere Analyse des Designraums identifiziert zk-SNARKs, eine Klasse nicht-interaktiver Protokolle für kryptographische Zero-Knowledge Beweise, als vielversprechenden Ansatz. Allerdings ist die Instanziierung dieser Protokolle komplex und somit wenigen Experten vorbehalten. Geeignete Programmierabstraktionen und softwaretechnische Werkzeuge fehlen. Um dieses Problem zu adressieren, präsentieren wir ZoKrates, die erste höhere Programmiersprache und Sammlung von Softwarewerkzeugen zur Übersetzung und Ausführung zk-SNARK-basierter verifizierbarer Off-Chain Berechnungen. Wir demonstrieren Relevanz und Anwendbarkeit an drei dezentralen Applikationen: Peer-to-Peer Energiehandel, Blockchain-Relays und anonyme Token-Transfers. Die im Kontext dieser Arbeit entstandenen Softwarelösungen finden darüber hinaus unabhängige Anwendung in Wissenschaft und Industrie.

1 Motivation und Problemstellung

Blockchaintechnologien erlauben sich gegenseitig misstrauenden Akteuren zensurresistent Transaktionen in einem verteilten System zu verarbeiten und dabei eine unveränderliche Transaktionshistorie zu etablieren, ohne hierfür eine vertrauenswürdige dritte Partei hinzuzuziehen. Allerdings stehen diese Eigenschaften mit anderen wünschenswerten Qualitätseigenschaften verteilter Systeme in Konflikt.

In aktuellen Blockchain-Netzwerken steigt der Durchsatz nicht mit der Anzahl der aktiven Knoten. Blockchains skalieren nicht. Der Durchsatz ist gering, die Transaktionskosten und Verarbeitungslatenzen sind hoch: Bitcoin verarbeitet derzeit 7 Transaktionen pro Sekunde, und Blöcke, die eine Reihe von Transaktionen bestätigen, werden im Durchschnitt alle 10 Minuten erstellt; Ethereum verarbeitet bis zu 25 Transaktionen pro Sekunde und hat ein durchschnittliches Blockintervall von 15 Sekunden. Im Vergleich dazu verarbeitet

[1] Englischer Titel der Dissertation [Eb21]: “Scalable and Privacy-preserving Off-Chain Computations”

[2] Die Dissertation ist in der Forschungsgruppe Information Systems Engineering (ISE) an der Technischen Universität Berlin entstanden. Kontakt: mail@jacobeberhardt.de

der Zahlungsabwickler Visa im Durchschnitt circa 1700 Transaktionen pro Sekunde, die innerhalb von Sekunden bestätigt werden. Dies ist ein grundlegender Nachteil für dezentrale Anwendungen, die mit traditionellen Diensten konkurrieren, die nicht unter solchen Einschränkungen leiden. Das Problem der Skalierbarkeit ist in der Forschung wohl bekannt und wird als intrinsisch schwierig erachtet [Cr16].

Die zweite wünschenswerte Eigenschaft ist der Schutz der Privatsphäre und die Möglichkeit zur Verarbeitung vertraulicher Daten. Diese Anforderung steht jedoch in einem grundlegenden Widerspruch zur derzeitigen Funktionsweise moderner Blockchains: In aktuellen Blockchain-Netzwerken führen alle Knoten redundant jede einzelne Transaktion aus. Diese Arbeitsweise ist grundsätzlich erforderlich, um die Korrektheit der Verarbeitungsergebnisse zu gewährleisten. Gleichzeitig bedeutet dies, dass alle Informationen, die verarbeitet werden, allen Knoten im Netzwerk bekannt sein und von ihnen gespeichert werden müssen. Andernfalls wäre eine redundante Ausführung nicht möglich. Folglich dürfen private oder vertrauliche Daten nicht auf der Blockchain verarbeitet werden - sie würden sofort netzwerköffentlich.

In der diesem Artikel zugrunde liegenden Arbeit widmen wir uns der Frage, wie diese grundlegenden Herausforderungen in Bezug auf Skalierbarkeit und Datenschutz in Blockchain-basierten Anwendungen adressiert werden können. Wir stellen unsere Beiträge und Resultate diesbezüglich in den nachfolgenden Abschnitten in verkürzter Form dar; für eine wesentlich tiefere Darstellung verweisen wir auf die Dissertationsschrift [Eb21]. In Abschnitt 2 führen wir zunächst Off-Chaining als grundlegenden Mechanismus, um Skalierungs- und Datenschutzprobleme dezentraler Anwendungen zu lösen, ein. Darauf aufbauend entwickeln wir in Abschnitt 3 ZoKrates, eine Programmiersprache und Sammlung von Softwarewerkzeugen, die es Entwicklern dezentraler Anwendungen ermöglicht, Off-Chain Berechnungen auf nutzerfreundliche Art zu spezifizieren und auszuführen. Im Rahmen einer ausführlichen Evaluation wird die praktische Signifikanz von ZoKrates und Off-Chaining in Abschnitt 4 durch die exemplarische Anwendung auf drei relevante Blockchain-basierte Applikationen demonstriert, welche sich mit Skalierbarkeits- oder Datenschutzproblemen konfrontiert sehen.

2 Off-Chaining und Off-Chain Berechnungen

In unserer Arbeit schlagen wir Off-Chaining als einen grundlegenden Ansatz vor, um Herausforderungen im Bezug auf Skalierbarkeit und Datenschutz im Kontext Blockchain-basierter Anwendungen zu adressieren.

Wir definieren Off-Chaining als die Auslagerung von Berechnungen und/oder Daten aus der Blockchain, wobei die wichtigsten Eigenschaften der Blockchain so wenig wie möglich beeinträchtigt werden. Die Kernidee besteht darin, die Datenspeicherung sowie den Rechenaufwand auf der Blockchain zu minimieren, indem Blockchain-externe Ressourcen, wie genutzt werden. Durch die Verringerung des Verarbeitungsaufwands auf der Blockchain

werden Kapazitäten für andere dezentrale Anwendungen frei. Außerdem ist die Speicherung sensibler Daten außerhalb der Blockchain die einzige Möglichkeit, die Privatsphäre zu gewährleisten — alle auf der Blockchain gespeicherten Informationen sind per Definition öffentlich einsehbar, da sie von allen Knoten zur Transaktionsvalidierung genutzt werden müssen.

2.1 Off-Chaining Entwurfsmuster

Um die Lücke zwischen dieser abstrakten Definition und praktikablen Off-Chaining-Ansätzen zu schließen, analysieren wir wiederkehrende Herausforderungen und Lösungsideen im dezentralen Anwendungsdesign [ET17]. In der Arbeit strukturieren und kategorisieren wir diese in fünf verschiedene Off-Chaining Entwurfsmuster bzw. Patterns:

1. Inhaltsadressierbarer Off-Chain Speicher Entwurfsmuster
2. Verifizierbare Off-Chain Berechnungen Entwurfsmuster
3. Off-Chain Signaturen Entwurfsmuster
4. Optimistische Finalisierungs Entwurfsmuster
5. Niedriger Contract Fußabdruck Entwurfsmuster

Die Instanziierung dieser Entwurfsmuster ermöglicht es Entwicklern, die Herausforderungen der Skalierbarkeit und des Datenschutzes zu bewältigen, mit denen sie bei der Entwicklung von Blockchain-basierten Anwendungen häufig konfrontiert werden. Während Skalierbarkeit und Datenschutz nicht im Widerspruch zueinander stehen und durch Off-Chaining gleichzeitig angegangen werden können, stellen wir fest, dass ein Tradeoff zur Verfügbarkeit besteht, der sorgfältige Abwägung verlangt.

2.2 Off-Chaining von Berechnungen

Aus unserer Analyse schließen wir, dass Off-Chain Berechnungen, wie sie im verifizierbare Off-Chain Berechnungen Entwurfsmuster beschrieben werden, besonders gut geeignet sind, um Datenschutzanforderungen in dezentralen Anwendungen zu begegnen, da sie eine vertraulichkeitswahrende Verarbeitung von Off-Chain Daten ermöglichen.

Wenn die Verifizierung von Off-Chain berechneten Ergebnissen auf der Blockhain außerdem kostengünstiger ist als die On-Chain Ausführung eben dieser Berechnung, kann dieser Ansatz den Durchsatz zudem direkt verbessern. Während in der Literatur einige Realisierungen von Off-Chain Berechnungen für bestimmte Kontexte vorgeschlagen wurden, gibt es keine systematische Analyse möglicher Ansätze, ihrer Eigenschaften und ihres Vergleichs.

In unserem zweiten Hauptbeitrag befassen wir uns mit diesem Problem, indem wir systematisch den Designraum für Off-Chain Berechnungen untersuchen, grundlegende Kategorien von Off-Chain Berechnungsansätzen identifizieren und bestehende Vorschläge aus der weißen und grauen Literatur kategorisieren. Anschließend führen wir eine vergleichende Analyse durch, bei der die Ansätze in Bezug auf Skalierbarkeit, Datenschutz, Sicherheit und Programmierbarkeit gegenübergestellt werden.

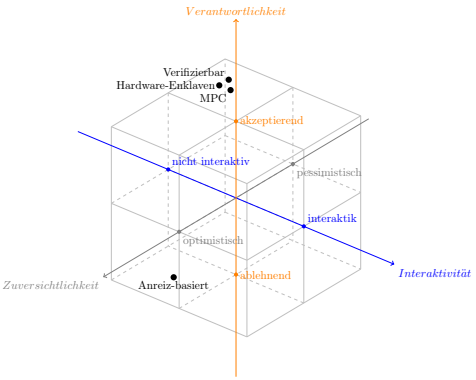


Abb. 1: Designraum der Protokolle für Off-Chain Berechnungen: Dimensionen und Ausprägungen.

Im Rahmen dieser Analyse wurden vier grundlegende Ansätze identifiziert, die sich in der Art, wie die Korrektheit der Blockchain-externen Berechnungen sichergestellt wird, unterscheiden [EH18]: Kryptographisch verifizierbare Berechnungen generieren direkt überprüfbare Korrekheitszertifikate, während Enklaven-basierte Ansätze sich auf isolierte Hardwaremodule verlassen. Anreizbasierte Ansätze nutzen interaktive Protokolle in Kombination mit werthaltigen Blockchain-Tokens, z. B. Bitcoin, um die Korrektheit von Berechnungsergebnissen spieltheoretisch durchzusetzen. Der letzte Ansatz basiert auf der Ausführung von Secure Multiparty Computation (MPC) Protokollen in einem Netzwerk von Off-Chain Knoten.

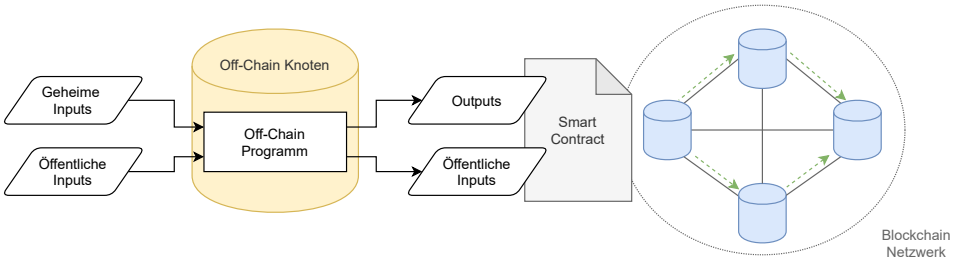


Abb. 2: Komponenten in Protokollen für Off-Chain Berechnungen: Ein Blockchain-externer Knoten erhält öffentliche und geheime Inputs, verarbeitet diese in einem Off-Chain Programm und sendet die Berechnungsergebnisse, sowie öffentliche Inputs an einen Smart Contract.

Als Ergebnis der vergleichenden Analyse stellen wir fest, dass zk-SNARKs aus der Gruppe der kryptographisch verifizierbaren Berechnungen einen besonders leistungsfähigen Ansatz im Bezug auf Skalierbarkeit und Datenschutz darstellen. Eine vereinfachte Ergebnisübersicht ist in Tab. 1 dargestellt.

Tab. 1: Vergleich der Ansätze für Off-Chain Berechnungen

		Skalierbarkeit		*Vertraulichkeit*	*Sicherheit*		*Programmierbarkeit*
Ansatz	**Realisierung**	**On-Chain Verifikation**	**Off-Chain Berechnung**	**Private Inputs**	**Sicherheitsannahme**	**Post-Quantum Sicherheit**	**Progr. Abstraktion**
Verifizierbare Berechnungen	zk-SNARK	*Einmaliges Setup:* $O(n)$, n Anzahl Multiplikationsgatter im Schaltkreis *Wiederholte Verifikation:* $O(1)$ *Beweisgröße:* $O(1)$, z.B. 3 Gruppenelemente [Gr16], i.e.127 bytes für BN254 Kurve	$O(n \log n)$, n Anzahl Multiplikationsgatter im Schaltkreis	ja	Knowledge of Exponent Annahme & Setup korrekt ausgeführt	nein	Arithmetische Schaltkreise
	Bulletproofs	*Verify:* $O(n)$, n Anzahl Multiplikationsgatter im Schaltkreis *Beweisgröße:* wenige Kilobytes, $O(\log n)$, n Anzahl Multiplikationsgatter im Schaltkreis	$O(n)$, n Anzahl Multiplikationsgatter im Schaltkreis	ja	Diskrete-Logarithmus-Annahme	nein	Arithmetische Schaltkreise
	zk-STARK	*Verify:* $O(\log^2 n)$, n Anzahl Multiplikationsgatter AIR zu Schaltkreis ausgerollt *Beweisgröße:* wenige hundert Kilobytes, $O(\log^2 n)$, n Anzahl Multiplikationsgatter AIR zu Schaltkreis ausgerollt	$O(n \log^2 n)$, n Anzahl Multiplikationsgatter AIR zu Schaltkreis ausgerollt	ja	Kollisionsresistente Hashfunktionen	ja	AIR (in Schaltkreise ausrollbar)
Hardware-Enklaven		Validierung der Attestation der Enklave: $O(1)$, Signaturprüfung	Native Ausführung & Attestations-Overhead	ja	TEEs sind isoliert & Vertrauen in Remote-Attestation Zertifikate	nein	Sprachen, die in TEE-kompat. Maschinencode kompilieren
Anreiz-basiert		Binärsuche & ein Berechnungsschritt: $O(\log n)$, n Anzahl Berechnungsschritte	Overhead Virtuelle Maschine (Ausführungshistorie)	nein	Ökonomisch rationale Teilnehmer	ja	Sprachen, die in VM-Instruktionsset kompilieren
MPC-basiert		*On-Chain Auditor:* $O(n)$, n Anzahl Gatter im Schaltkreis *Größe Audit-Trail:* $O(n)$, n Anzahl Gatter im Schaltkreis	$O(n)$, n Anzahl Gatter im Schaltkreis	ja	Mindestens ein ehrlicher Knoten & Mindestehrlichkeitsschranke für Schutz privater Inputs und Liveness	ja	Boolsche oder arithmetische Schaltkreise

3 ZoKrates - Programmierung von Off-Chain Berechnungen

In der vorangegangenen Analyse wurden zk-SNARKs als geeigneter Ansatz für die Realisierung von allgemeinen Off-Chain Berechnungen identifiziert. Allerdings ist die konkrete Instanziierung schwierig: Berechnungen müssen in schwer zu verwendenden Low-Level Abstraktionen spezifiziert werden, und die On-Chain Verifikation ist komplex, da sie tiefes Wissen über die verwendeten kryptographischen Protokolle erfordert.

Wir schließen diese Lücke mit dieser Arbeit, indem wir ZoKrates, das erste Framework für effiziente Zero-Knowledge Off-Chain Berechnungen entwerfen, implementieren und evaluieren [ET18]. ZoKrates ermöglicht es dezentralen Anwendungen, ihre Anforderungen an Datenschutz und Skalierbarkeit zu erfüllen, indem es eine entwicklerfreundliche Abstraktion für die Spezifikation, die Off-Chain Ausführung und die On-Chain Überprüfung von verifizierbaren Off-Chain Berechnungen auf Basis von zk-SNARKs bereitstellt. Benutzerfreundlichkeit, Effizienz und Allgemeingültigkeit stellten die Hauptziele für ZoKrates als Framework für verifizierbare Zero-Knowledge Off-Chain Berechnungen dar.

ZoKrates besteht aus einer domänenspezifischen Programmiersprache, die die Besonderheiten der zugrunde liegenden Abstraktionen abbildet und es Entwicklern ermöglicht, Off-Chain Berechnungen bequem als Programme auf Abstraktionsebene einer Hochsprache zu spezifizieren. Diese Programme werden dann in die proprietäre ZoKrates Intermediate Representation übersetzt und durch den ZoKrates Interpreter ausgeführt. Anschließend kann ein Korrektheitsbeweis für diese Programmausführung generiert werden. Um eine On-Chain Verifikation zu ermöglichen, unterstützt ZoKrates die Generierung und den Export von Verifikations-Smart Contracts, die die Korrektheit von Off-Chain Berechnungen überprüfen. In Abb. 3 geben wir einen Überblick über alle Schritte, die von der Spezifikation einer Berechnung als ZoKrates Programm bis zur Verifizierung dessen Ausführung auf der Blockchain erforderlich sind.

```
1  import "hashes/sha256/512bit" as sha256
2
3  def main(private u32[16] input) -> u32[8]:
4    u32[8] h = sha256(input[0..8], input[8..16])
5    return h
```

List. 1: Beispiel für ein einfaches ZoKrates Programm, das einen SHA-256-Hash auf geheimen Inputparametern berechnet. Damit kann das Wissen über das Urbild des errechneten SHA-256-Hashes bewiesen werden, ohne das Urbild je offenzulegen.

Die Implementierung des ZoKrates Frameworks ist seit der Veröffentlichung der ursprünglichen ZoKrates-Publikation [ET18] beträchtlich gereift und hat sich zu einem aktiven Open-Source-Projekt mit mehreren Beitragenden entwickelt. Dennoch haben sich die Kernkomponenten und Ideen nicht verändert. Code und Nutzerdokumentation sind auf GitHub verfügbar[3].

Während sich die Implementierung auf die Verwendung mit der Ethereum Blockchain fokussiert, unterstützt ihre Architektur jedoch beliebige Blockchains, die über eine ausreichend leistungsfähige Ausführungsumgebung für die Beweisverifizierung verfügen. Darüber hinaus wird jede Implementierung eines verifizierbaren Berechnungsschemas durch den

[3] https://github.com/ZoKrates/ZoKrates
[4] https://remix.ethereum.org

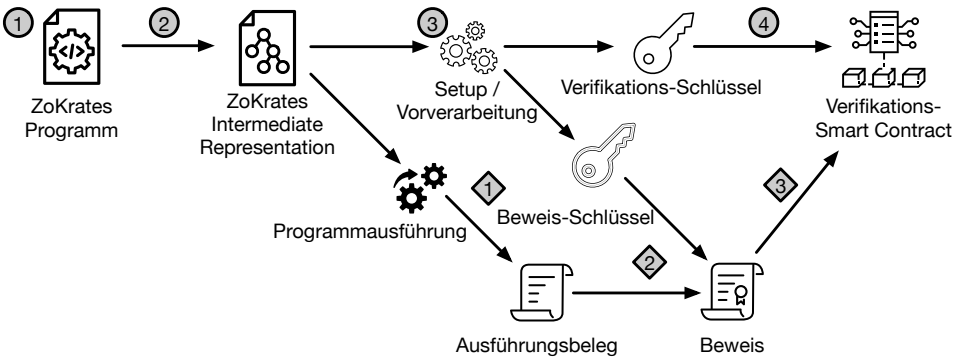


Abb. 3: Überblick über den ZoKrates Übersetzungs-, Ausführungs- und Beweisprozess. Zunächst wird eine Reihe von einmaligen Vorbereitungsschritten für ein Off-Chain Programm durchgeführt: Programmspezifikation, Kompilierung, Setup und Erzeugung eines Verifikations-Smart-Contracts. Diese Schritte sind mit eingekreisten Zahlen markiert. Anschließend wird das Off-Chain Programm ausgeführt, ein Beweis über die Korrektheit der Ausführung generiert und zur Überprüfung an den Verifikations-Smart-Contract übergeben. Diese Schritte werden für jede Programmausführung wiederholt und sind mit Rauten markiert.

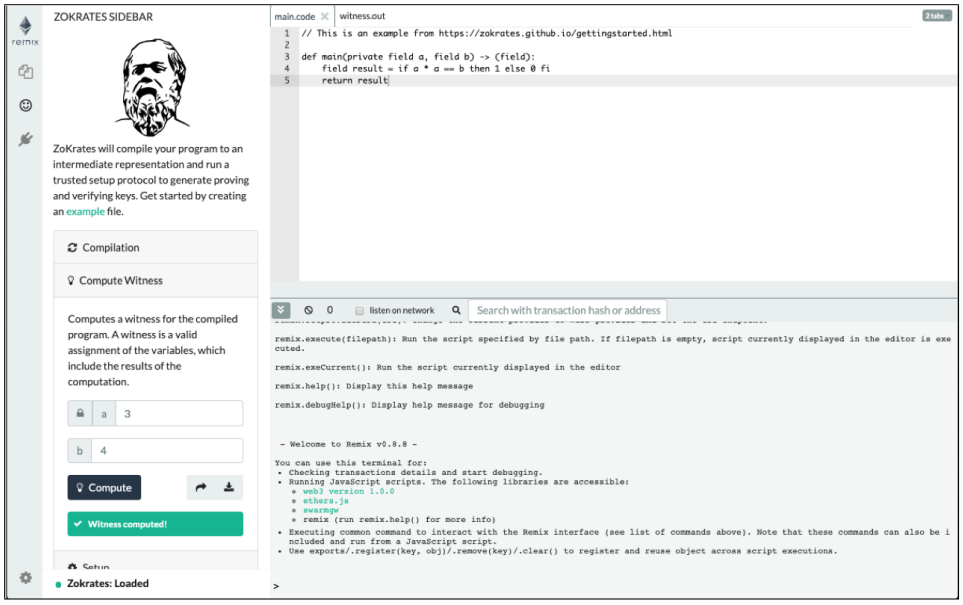


Abb. 4: ZoKrates Development in der Ethereum Remix IDE. Der Nutzer kann den kompletten Prozess von Programmspezifikation bis Beweisgenerierung und -verifizierung im Browser durchlaufen[4]

modularen Aufbau der Architektur unterstützt, solange das implementierte Schema mit der ZoKrates Intermediate Representation kompatibel ist.

Effizienz und Praktikabilität des grundlegenden Konzepts sowie der konkreten ZoKrates Implementierung werden in der zugrunde liegenden Arbeit in einer ausführlichen Performance-Evaluation belegt. Hierzu werden Programme aus den im nachfolgenden Abschnitt beschriebenen Anwendungen herangezogen.

4 Anwendungen

In einer ausführlichen Evaluierung zeigen wir, wie Datenschutz- und Skalierbarkeitsprobleme, mit denen reale Blockchain-basierte Anwendungen konfrontiert sind, durch verifizierbare Off-Chain Berechnungen gelöst werden können. Konkret beschreiben wir ZoKrates-basierte Varianten dezentraler Anwendungen für datenschutzfreundliche Token-Transfers und Peer-to-Peer-Energiehandel sowie ein skalierbares Blockchain-Relay und demonstrieren damit die Praxistauglichkeit des Frameworks.

Die dezentrale Peer-to-Peer-Energiehandelsanwendung und das skalierbare Blockchain-Relay wurden von uns mit unseren Co-Autoren vorgeschlagen, implementiert und eva-

luiert [Eb20]. Im Gegensatz dazu wurde die ZoKrates-basierte anonyme Token-Transfer-Anwendung unabhängig von der Blockchain Forschungs- und Entwicklungsabteilung von Ernst & Young entwickelt und implementiert, was Nutzbarkeit, Nützlichkeit und Reife von ZoKrates unterstreicht.

4.1 Privatsphäre-wahrender Energiehandel in zukünftigen Stromnetzen

In unserer ersten Anwendung verwenden wir ZoKrates, um Smart-Meter-Daten in zukünftigen Energienetzwerken zu verbergen und gleichzeitig eine vertrauenswürdige Verarbeitung zum Zweck der gemeinsamen Nutzung von Energie in einer Gemeinschaft von Haushalten zu ermöglichen. Wir haben unsere Lösung im Rahmen von BloGPV[5], einem nationalen Forschungsprojekt, implementiert und evaluiert [Eb20, WE20]. Das entstandene System schützt die Privatsphäre der teilnehmenden Personen und erhöht gleichzeitig die Rentabilität der erneuerbaren Energieerzeugung.

Ganz allgemein zeigen wir, wie ZoKrates-basierte Off-Chain Berechnungen mit On-Chain Commitments kombiniert werden können, um Algorithmen in einer sich mistrauenden Gruppe mit Blockchain-Eigenschaften auszuführen und dabei die Privatsphäre zu wahren.

4.2 zkRelay

Zweitens stellen wir zkRelay vor, ein skalierbares Blockchain-Relay, das Off-Chain Berechnungen für die Validierung von Block-Headern nutzt [WE20]. Wir demonstrieren, wie ZoKrates-basierte Off-Chain Berechnungen verwendet werden können, um einer Blockchain zu ermöglichen, Daten und Ereignisse einer anderen Blockchain auf effiziente und skalierbare Weise zu validieren. Wir stellen ein Relay-Design vor, das die Header-Validierung durch überprüfbare Off-Chain Berechnungen außerhalb der Blockchain realisiert und dadurch die Kosten für die Validierung von Blockheadern einer Quell-Blockchain auf einer Ziel-Blockchain reduziert. Als Proof-of-Concept stellen wir eine ZoKrates-basierte Implementierung für ein Bitcoin-Relay auf der Ethereum-Blockchain zur Verfügung, die die Validierung von 504 Bitcoin-Headern in einer einzigen Ethereum-Transaktion ermöglicht.

Unsere zkRelay-Implementierung reduziert die Kosten für die Validierung von Bitcoin-Headern auf der Ethereum-Blockchain um das bis zu 187-fache im Vergleich zu BTC-Relay, der state-of-the-art Lösung.

4.3 Anonyme Tokentransfers

Neben protokollnativen Tokens, z. B. Bitcoin oder Ether, ermöglichen programmierbare Blockchain-Plattformen den Entwicklern dezentraler Anwendungen, ihre eigenen Token

[5] `https://blogpv.net/`

durch Smart Contracts zu erstellen. Solche Token implementieren häufig eine standardisierte Schnittstelle, um die Kompatibilität mit Börsen und anderen Anwendungen zu gewährleisten, z. B. den ERC-20-Standard. Wie native Tokens leiden auch diese benutzerdefinierten Tokens unter schwachen Datenschutzgarantien: Eigentumsinformationen werden einsehbar in Smart Contracts gespeichert, bei jeder Übertragung sind Absender und Empfänger sowie die Anzahl der übertragenen Token für alle Netzwerkteilnehmer sichtbar. Pseudonyme Adressen bieten keinen ausreichenden Schutz der Privatsphäre [An13, RH13].

Um dieses Problem zu adressieren, hat Ernst & Young Nightfall entwickelt, ein Protokoll, das datenschutzkonforme Token-Transfers für das öffentliche Ethereum-Netzwerk realisiert. Hierzu erweitert Nightfall die Ideen von Zerocash [Sa14], um datenschutzfreundliche Übertragungen von Ethereum-basierten benutzerdefinierten Tokens nach den Standards ERC-20 und ERC-721 zu unterstützen. ZoKrates dient hierbei als zentrales Werkzeug zur Realisierung der Protokollprimitive: Es erlaubt die Einhaltung von Transferregeln in Off-Chain Berechnungen zu beweisen ohne die dabei verwendeten Daten auf der Blockchain zu veröffentlichen. Token-Übertragungen in Nightfall sind immer anonym, d.h., Sender und Empfänger bleiben verborgen.

5 Zusammenfassung

In der diesem Artikel zu Grunde liegenden Dissertation wurde Off-chaining als grundlegender Ansatz eingeführt, um Skalierbarkeits- und Datenschutzprobleme im Kontext Blockchain-basierter Anwendungen zu adressieren. Off-Chaining-basierte Lösungsideen für wiederkehrende Herausforderungen im Design dezentraler Anwendungen wurden identifiziert und in Form von Entwurfsmuster strukturiert. Eine vergleichende Analyse von Instanziierungsoptionen für Off-chain Berechnungen zeigte die besondere Eignung von zk-SNARKs, einer Klasse nicht-interaktiver kryptographischer Protokolle für Zero-Knowledge Beweise. Es fehlten jedoch geeignete Programmierabstraktionen, der Einsatz bleibt Experten vorbehalten.

Diese Lücke schließen wir mit ZoKrates, einer höheren Programmiersprache und Sammlung von Softwarewerkzeugen zur Übersetzung und Ausführung zk-SNARK-basierter verifizierbarer Off-Chain Berechnungen. Im Rahmen einer ausführlichen Evaluation demonstrierten wir die Praktikabilität und Anwendbarkeit von ZoKrates an drei dezentralen Applikationen: dezentraler Energiehandel, Blockchain-Relays und anonyme Token-Transfer. Über die Arbeit hinausgehend finden ZoKrates und die zugehörigen Softwarewerkzeuge unabhängige Anwendung in Wissenschaft und Industrie.

Literaturverzeichnis

[An13] Androulaki, Elli; Karame, Ghassan O; Roeschlin, Marc; Scherer, Tobias; Capkun, Srdjan: Evaluating user privacy in bitcoin. In: International Conference on Financial Cryptography and Data Security. Springer, S. 34–51, 2013.

[Cr16] Croman, Kyle; Decker, Christian; Eyal, Ittay; Gencer, Adem Efe; Juels, Ari; Kosba, Ahmed; Miller, Andrew; Saxena, Prateek; Shi, Elaine; Sirer, Emin Gün et al.: On scaling decentralized blockchains. In: International Conference on Financial Cryptography and Data Security. Springer, S. 106–125, 2016.

[Eb20] Eberhardt, Jacob; Peise, Marco; Kim, Dong-Ha; Tai, Stefan: Privacy-Preserving Netting in Local Energy Grids. In: Proceedings of the IEEE International Conference on Blockchain and Cryptocurrency 2020. IEEE, 2020.

[Eb21] Eberhardt, Jacob: Scalable and privacy-preserving off-chain computations. Doctoral thesis, Technische Universität Berlin, 2021.

[EH18] Eberhardt, Jacob; Heiss, Jonathan: Off-chaining Models and Approaches to Off-chain Computations. In: Proceedings of the 2nd Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers. ACM, S. 7–12, 2018.

[ET17] Eberhardt, Jacob; Tai, Stefan: On or Off the Blockchain? Insights on Off-Chaining Computation and Data. In: Proceedings of the European Conference on Service-Oriented and Cloud Computing. Springer, S. 3–15, 2017.

[ET18] Eberhardt, Jacob; Tai, Stefan: ZoKrates - Scalable Privacy-Preserving Off-Chain Computations. In: IEEE International Conference on Blockchain. IEEE, 2018.

[Gr16] Groth, Jens: On the size of pairing-based non-interactive arguments. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, S. 305–326, 2016.

[RH13] Reid, Fergal; Harrigan, Martin: An analysis of anonymity in the bitcoin system. In: Security and Privacy in Social Networks, S. 197–223. Springer, 2013.

[Sa14] Sasson, Eli Ben; Chiesa, Alessandro; Garman, Christina; Green, Matthew; Miers, Ian; Tromer, Eran; Virza, Madars: Zerocash: Decentralized anonymous payments from bitcoin. In: Security and Privacy (SP), 2014 IEEE Symposium on. IEEE, S. 459–474, 2014.

[WE20] Westerkamp, Martin; Eberhardt, Jacob: zkRelay: Facilitating Sidechains using zkSNARK-based Chain-Relays. In: Proceedings of the IEEE European Symposium on Security and Privacy Workshops. IEEE, S. 378–386, 2020.

Jacob Eberhardt promovierte 2021 in der Gruppe Information Systems Engineering von Prof. Tai an der Technischen Universtität Berlin. Seine Forschungsarbeiten wurden mit einem IEEE Best Paper Award ausgezeichnet, gewannen einen Samsung Next Research Grant und resultierten in einem Open Source Projekt mit Förderung durch die Ethereum Foundation. Zudem diente er in verschiedenen Programmkomitees internationaler Konferenzen im Bereich Blockchains. Zuvor legte er einen Bachelor und Masterabschluss in Wirtschaftsingenieurwesen mit Schwerpunkt Informatik am Karlsruher Institut für Technologie (KIT) ab.

Selbstadaptive Fitness in evolutionären Prozessen[1]

Thomas Gabor[2]

Abstract: Evolutionäre Prozesse modellieren die Entwicklung von Objekten mit zunächst zufälligen Eigenschaften zu Objekten, deren Eigenschaften einer Ordnung oder einem bestimmten Ziel (genannt *Fitness*) folgen. Evolutionäre Prozesse treten in Software häufig als Optimierungsalgorithmen oder beim maschinellen Lernen auf, wobei ihr Ziel meist extrinsisch durch einen Designer oder Programmierer bestimmt ist. Oft ist es jedoch von Vorteil, wenn besagte Algorithmen ihre Fitnessberechnung während ihrer Ausführung intrinsisch selbst adaptieren können. Wir verfolgen dieses Phänomen zurück auf künstliche Chemiesysteme (*artificial chemistry systems*), wo Fitness ohne Designer entsteht. Wir untersuchen diversitätsbasierte Fitnessfunktionen in evolutionären Algorithmen und können erstmalig ihre Effektivität begründen, indem wir das theoretische Modell der produktiven Fitness definieren. Schließlich finden wir einen Effektivitätsgewinn auch beim Zusammenspiel von evolutionären Algorithmen und bestärkendem Lernen (*reinforcement learning*), wobei beide Methoden allein durch eine wechselseitig adaptive Fitness interagieren. Dieses Konzept lässt sich auch als Architekturmuster für Softwaresysteme verallgemeinern.

1 Einleitung

Evolutionäres Rechnen (*evolutionary computing*) hat sich von grundlegenden Konzepten der chemischen oder biologischen Evolution inspirieren lassen, um leistungsfähige und interessante Algorithmen zu entwerfen. Im praktischen Einsatz bleibt oft ein frappierender Unterschied zwischen der in Software umgesetzten Evolution und ihrem natürlichen Vorbild: der Ursprung der Fitness. In der biologischen Evolution geht der Begriff der *Fitness* auf Charles Darwin [Da09] selbst zurück und beschreibt ein abstraktes Konzept zur Erklärung *a posteriori*, bspw. warum sich bestimmte Merkmale eher durchsetzen als andere. Im Kontext des evolutionären Rechnens in Computern hat Fitness dagegen meist einen stark imperativen Charakter, denn über eine Definition von Fitness in Software teilt der Programmierer dem Algorithmus *a priori* die zu erreichenden Ziele mit [De17].

Selbst-adaptive Fitness vereint diese beiden Perspektiven: Eine Zielfunktion kann extrinsisch vorgegeben sein, doch sie bzw. ihre Auswirkung auf die Fitness wird aus dem Ablauf der Evolution heraus, also intrinsisch, angepasst. Formal definieren wir zunächst einen evolutionären Prozess, der an einer gegebenen (objektiven) Zielfunktion t gemessen wird. O.B.d.A. nehmen wir an, dass die Werte der Zielfunktion im Bereich $[0; 1]$ liegen und wir deren Maximierung verfolgen.

[1] Englischer Titel der Dissertation: "Self-Adaptive Fitness in Evolutionary Processes"

[2] Ludwig-Maximilians-Universität, Fakultät für Mathematik, Informatik und Statistik, Institut für Informatik, Oettingenstraße 67, 80538 München, Deutschland
thomas.gabor@ifi.lmu.de

Definition 1 (Evolutionärer Prozess) *Sei $\mathcal{X}$ ein beliebiger Suchraum (bspw. $\mathbb{R}^{42}$). Eine potenziell randomisierte Funktion*[3] *$e : \wp(\mathcal{X}) \to \wp(\mathcal{X})$ heiße evolutionäre Schrittfunktion. Eine Funktion $t : \mathcal{X} \to [0;1]$ heiße Zielfunktion. Sei $\langle X_i \rangle_{1 \leq i \leq n}$ eine Serie von sogenannten Populationen, d.h. $X_i \subset \mathcal{X}$ für alle i. Sei $X_0 \subset \mathcal{X}$ eine initiale Population. Ein Tupel $\mathcal{E} = (\mathcal{X}, E, t, \langle X_i \rangle_{i \leq g})$ heißt* evolutionärer Prozess *in g-ter Generation gdw. $X_i = e(X_{i-1})$ für alle $i \leq g$. Ein evolutionärer Prozess entwickelt sich gut, wenn $t(X_j) > t(X_i)$ für viele*[4] *i und jeweils ein $j > i$.*

Um sich gut zu entwickeln benötigt die evolutionäre Schrittfunktion e meist selbst Kenntnis über zumindest das ungefähre Ziel des evolutionären Prozesses. Dies erfolgt durch eine Fitnessfunktion $f : \mathcal{X} \times \wp(\mathcal{X}) \to [0;1]$, die ähnlich wie eine Zielfunktion zu verstehen ist, aber den aktuellen Zustand des evolutionären Prozesses (in unserem Fall üblicherweise die Population der aktuellen Generation) in eine Wertabschätzung miteinbeziehen kann. Natürlich können wir schlicht $f(x, \_) = t(x)$ wählen, doch wir zeigen im Laufe dieses Textes, dass eine Fitness auch ohne eine Zielfunktion entstehen kann (Abschnitt 2), die ideale Fitnessfunktion zum Maximieren von t praktisch und theoretisch nicht exakt t selbst ist (Abschnitt 3) und sich hilfreiche Fitnessfunktion automatisiert generieren lassen (Abschnitt 4). Damit orientieren wir uns an den drei Kernkapiteln 3 – 5 der diesem Text zugrundeliegenden Doktorarbeit [Ga21a].

2 Die Entstehung von Fitness

Wir betrachten zunächst künstliche Chemiesysteme (*artificial chemistry systems*). Diese bestehen aus mehreren Partikeln, die meist mit zufälligen Eigenschaften initialisiert sind, jedoch fixen Interaktionsregeln folgen, durch die für eine Menge an Partikeln ein bestimmtes Verhalten oder eine bestimmte Struktur entstehen kann. Derartige System sind gut untersucht, wenn man bspw. Skalare, Automaten oder λ-Ausdrücke als Partikel annimmt und entsprechende Interaktionen wie respektive mathematische Operationen, Automatenvereinigungen oder λ-Applikation als Interaktionen definiert [BY15; DZB01; FB96]. In den jeweiligen Experimenten ist oft zu beobachten, dass die Menge von Partikeln (in diesem Kontext auch Suppe genannt) bestimmte Zielzustände anzustreben scheint, auch wenn der Versuchsaufbau keine extrinsische Belohnung für bestimmte Zustände vorsieht. Dennoch scheinen bestimmte Interaktionsregeln Partikel dahingehend zu beeinflussen, dass sie stabilere oder sich selbst replizierende Belegungen ihrer Eigenschaften anstreben. Diesem Phänomen wird in der chemischen Ursuppe nicht zuletzt auch die Entstehung von Leben auf dem Planeten Erde zugeschrieben [Da96].

[3] Wir verwenden den Begriff *Funktion* in diesem Text eher im Sinne üblicher (imperativer) Programmiersprachen. Unsere Funktionen können dabei für dieselben Inputs unterschiedliche Outputs liefern. Für eine genauere mathematische Betrachtung verweisen wir auf den Ursprungstext [Ga21a].

[4] Die genaue Definition des Zielkriteriums hängt von der praktischen Umsetzung des Prozesses und seiner Anwendung ab. Für diese generische Definition bleiben wir bewusst unspezifisch und verweisen auf den Ursprungstext [Ga21a].

Diese „natürlichen Zielzustände“ des künstlichen Chemiesystems lassen sich als eine natürlich entstandene, emergente Fitnessfunktion auffassen. Da diese allein aus der Evolution einer Suppe entsteht und stark von den aktuell in der Suppe lebenden Partikeln abhängt, erfüllt eine derartige Fitness auch unsere Definition von Selbst-Adaption. Gleichzeitig können wir durch deren “natürliche” Entstehung zeigen, dass evolutionäre Prozesse eine intrinsische Fitness jenseits ihrer gegebenen Ziele entwickeln können. Wir werden diese Fitness später (siehe Kapitel 3, 4) auch in Ergänzung zu einer extrinischen Fitness beobachten.

Im Rahmen dieser Doktorarbeit wurden künstliche Chemiesysteme entwickelt, deren Partikel neuronale Netze sind [Ga19a]. Ähnlich wie oben erwähnte Automaten oder λ-Ausdrücke können neuronale Netze komplexe Funktionen repräsentieren. Ein neuronales Netz $\mathcal{N}$ führt eine Funktion $\mathcal{N} : \mathbb{R}^p \to \mathbb{R}^q$ aus und ist dabei definiert durch einen Gewichtsvektor $\overline{\mathcal{N}} \in \mathbb{R}^n$ aus $|\overline{\mathcal{N}}| = n$ reellen Zahlen [Kr11]. Da ein neuronales Netz stets mehr Gewichte als Inputs aufweist, d.h. $|\overline{\mathcal{N}}| > p$, lässt sich ein neuronales Netz nicht trivialerweise auf ein anderes neuronales Netz derselben Größe (bzgl. $p, q, n \in \mathbb{N}$) anwenden. Wir untersuchen daher mehrere *Reduktionen*, die es ermöglichen den Informationsgehalt des Netzes zu vermindern oder ein Netz iterativ in Teilen einem anderen Netz als Input weiterzugeben [CL18; Ga19a]. Wir schreiben hier kurz $\mathcal{M}(\overline{\mathcal{N}})$ für die Anwendung von $\mathcal{M}$ auf die Gewichte von $\mathcal{N}$, auch wenn wir dabei insbesondere für den Fall $|\overline{\mathcal{M}}| = |\overline{\mathcal{N}}|$ erwähnte Reduktionen anwenden müssen. Dies erlaubt uns die Einführung folgender zwei (Inter-)Aktionen:

Self-Train. Die Aktion *self-train*$(\mathcal{N})$ mit Hyperparameter $A \in \mathbb{N}$ für die Intensität des Trainings trainiert mittels Backpropagation [Kr11] ein einzelnes neuronales Netz $\mathcal{N}$ darauf, seine eigenen Gewichte wiederzugeben, d.h. $\mathcal{N}(\overline{\mathcal{N}}) = \overline{\mathcal{N}}$ anzunähern. Dabei wird jedoch nur die aktuelle Gewichtskonfiguration von $\mathcal{N}$ zum Training benutzt und es werden nicht bspw. zufällige Vektoren aus dem Inputraum gezogen.

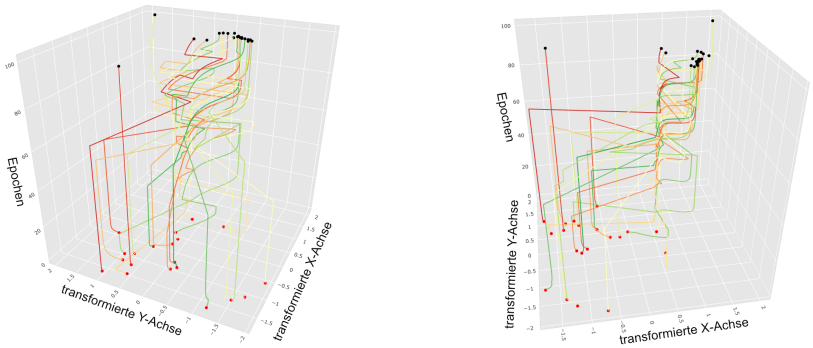


Abb. 1: Zwei Perspektiven auf die Evolution einer Suppe bestehend aus 20 zufällig initialisierten neuronalen Netzen mit jeweils zwei versteckten Schichten mit jeweils zwei Zellen. Die 20 Gewichte pro Netzwerk werden mittels *Principal Component Analysis (PCA)* in zwei Dimensionen X und Y dargestellt. Bild aus [Ga19a].

Attack. Die Interaktion *attack*$(\mathcal{M}, \mathcal{N})$ ersetzt die Gewichte des neuronalen Netzes $\mathcal{M}$ durch das Ergebnis der Anwendung von $\mathcal{N}$ auf diese Gewichte, setzt also $\overline{\mathcal{M}} := \mathcal{N}(\overline{\mathcal{M}})$.

Als ein Beispiel definieren wir nun eine Suppe, in der in jedem Evolutionsschritt (auch *Epoche* genannt) jeder Partikel der Aktion *self-train* mit Intensität $A = 30$ ausgesetzt wird und jeder Partikel mit einer Wahrscheinlichkeit von 0.1 einen zufälligen anderen Partikel für eine Interaktion *attack* zugewiesen bekommt. Abbildung 1 zeigt die entstehende Evolution. Hier können wir beobachten, dass auch ohne jede Zielvorgabe die Partikel zu einem bestimmten Bereich ihres Datenraums tendieren. Der ausgewählte Bereich ist je nach Initialisierung und anderen Zufallsfaktoren unterschiedlich. Meist bringt die Suppe jedoch vornehmlich Partikel hervor, die wir ε-Fixpunkte nennen.

Definition 2 (ε-Fixpoint) *Ein neuronales Netz $\mathcal{N}$ wird ε-Fixpunkt genannt, wenn die Anwendung des Netzes auf seine eigenen Gewichte die ursprünglichen Gewichte mit einem Fehler von höchstens $\varepsilon \in \mathbb{R}$ pro Element des Gewichtsvektors wiedergibt, d.h. $|\overline{\mathcal{N}(\overline{\mathcal{M}})}_i - \overline{\mathcal{N}}_i| < \varepsilon$ für jedes Gewicht mit Index i innerhalb der jeweiligen Netze.*

Ähnlich wie für Automaten oder λ-Ausdrücke bereits entdeckt, entsteht in dieser Art von Suppe eine Tendenz zu Stabilität und Selbst-Replikation, die man als emergente, intrinsische Fitnessfunktion auffassen kann. Mit neuronalen Suppen haben wir dabei ein spannendes Werkzeug geschaffen, um komplexe Algorithmen als Partikel mit (vordergründig) konstanter Platzkomplexität zu kodieren [Ga19a; Ga21b].

3 Die ideale Fitness

Wir folgen nun der Annahme, dass möglicherweise jeder evolutionäre Prozess eine intrinsische Fitness bzw. Fitnesskomponente mitbringt, auch wenn wir ihn extrinsisch in eine möglicherweise gänzlich andere Richtung steuern wollen. Zahlreiche Arbeiten beobachten ein Phänomen, dass diese Annahme unterstützt: Wenn wir eigentlich Individuen für ein bestimmtes Ziel gegeben durch eine Zielfunktion t optimieren wollen, können wir t dem evolutionären Prozess direkt als Fitnessfunktion f übergeben, also $f := t$; oft finden sich jedoch andere Fitnessfunktionen $f' \neq t$, die das Ziel t aber besser optimieren. Dieses Problem wird oft als *fitness design* oder (vor allem im Kontext von verstärkendem Lernen) als *reward engineering* bezeichnet. Wir untersuchen in diesem Abschnitt beispielhaft evolutionäre Algorithmen.

Definition 3 (Evolutionärer Algorithmus) *Sei mut : $\wp(X) \to \wp(X)$ eine Funktion, die aus einzelnen Individuen einer gegebenen Population neue (leicht veränderte) Varianten generiert, rec : $\wp(X) \to \wp(X)$ eine Funktion, die aus mehreren Individuen einer gegeben Population neue Individuen mit kombinierten Eigenschaften erzeugt, mig : $() \to \wp(X)$*

eine Funktion, die eine bestimmte Menge an Individuen zufällig generiert, und $sel : \mathbb{N} \times \wp(X) \to \wp(X)$ *eine Funktion, die eine gegebene Anzahl an Individuen aus einer gegebenen Population (probabilistisch und mit Bevorzugung nach einer Fitness* f*) auswählt. Ein evolutionärer Prozess* $\mathcal{E} = (X, e, t, \langle X_i \rangle_{i \leq g})$ *heißt* evolutionärer Algorithmus *gdw.* e *die Gestalt* $e(X) = sel(|X|, X \cup mut(X) \cup rec(X) \cup mig())$ *hat.*

Ein häufig implementiertes intrinsisches Ziel für evolutionäre Algorithmen ist Diversität. Im einfachsten Fall wird dabei eine Diversitätsfunktion $d : X \times \wp(X) \to \mathbb{R}$ vorausgesetzt, die die Diversität eines Individuums x in der Population X misst. Mit deren Hilfe kann eine neue diversitätsbewusste Fitnessfunktion $f'(x, X) = (1 - \zeta) \cdot t(x) + \zeta \cdot d(x, X)$ definiert werden. Wir sehen sofort, dass diese Fitnessfunktion selbst-adaptiv ist, da ihr Wert für ein gegebenes Individuum x von dem Zustand der Population X abhängt. Wineberg; Oppacher [WO03] zeigen, dass bspw. die durchschnittliche paarweise Manhattan-Distanz hier alle geläufigen Diversitätsfunktionen vertreten kann; wir zeigen, dass sich diese Distanz auch gut durch die paarweise Distanz zu nur einer kleinen zufälligen Untermenge der Gesamtpopulation gut approximieren lässt [Ga18; GB17; GBL18]. Wir entwickeln außerdem die Metrik der genealogischen Diversität, die die Diversität von Individuen anhand eines zufällig generierten Bitstrings ablesen lässt, der für zwei gegebene Individuen probabilistische Rückschlüsse über ihren Verwandtschaftsgrad zulässt [GB17]. Damit erzielen wir im Gegensatz zur Manhattan-Distanz zwar nicht notwendigerweise bessere Ergebnisse, doch wir benötigen keine Distanzfunktion zwischen den Daten der einzelnen Individuen, was praktisch wird, wenn diese keine reellwertigen Vektoren sondern bspw. Programmbäume oder neuronale Netze sind.

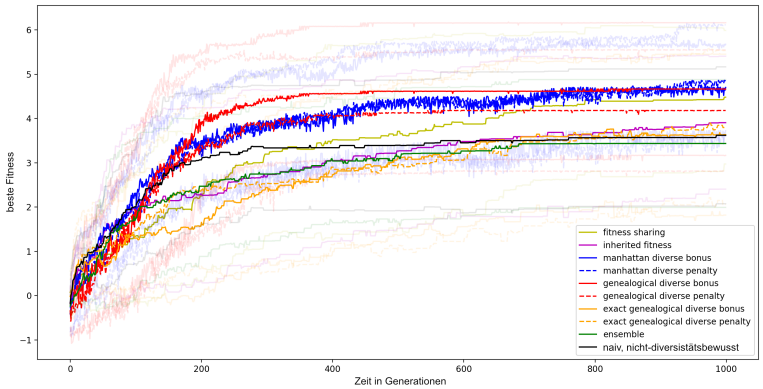


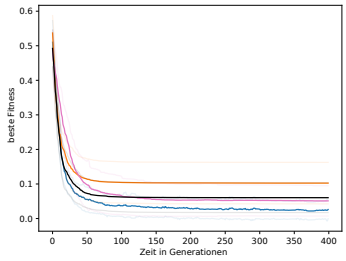
Abb. 2: Evaluation von evolutionären Algorithmen mit verschiedenen (meist diversitätsbasierten) Fitnessfunktionen für ein beispielhaftes Optimierungsproblem („Pathfinding“ [GBL18]). Es wurden jeweils 20 Durchläufe ausgeführt; ausgefüllte Linien zeigen den Durchschnitt, transparente Linien das Band einer Standardabweichung. Bild aus [GBL18].

Abbildung 2 zeigt ein Beispiel für die Optimierung derselben Zielfunktion mit Hilfe verschiedener Fitnessfunktionen. Wir sehen einen deutlichen Vorteil für (einige) diversitätsbasierte Fitnessfunktionen gegenüber der naiven Variante $f(x, \_) = t(x)$ und das eben obwohl diese Fitnessfunktionen das eigentliche Ziel zunächst zu verfälschen scheinen. Um dieses Phänomen erklären zu können, entwickeln wir das Modell der *final produktiven Fitness*, mit dem wir u.A. für bereits abgelaufene Evolutionen sagen können, welche Fitness bestimmte Individuen am besten hätten bekommen sollen.

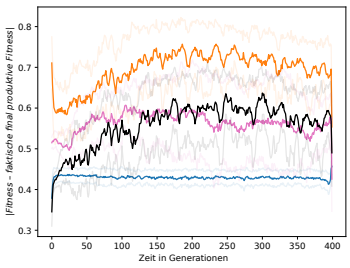
Definition 4 (Faktische final produktive Fitness) *Sei x ein gegebenes Individuum und X_i für $i = 1, ..., n$ eine Population der Generation i, so dass X_0 die zufällige Initialisierung eines evolutionären Algorithmus ist und X_i durch Evolution aus X_{i-1} hervorgeht. X_n ist die finale Population vor Abbruch der Evolution. D_x sei die Menge aller evolutionären Nachkommen des Individuums x. ω sei ein Schlimmstwert für die Zielfunktion t. Dann ist die final produktive Fitness $\phi^\dagger$ gegeben durch*

$$\phi^\dagger(x) = \begin{cases} \text{avg}_{x' \in D_x \cap X_n} \quad t(x') & \text{wenn } D_x \cap X_n \neq \emptyset \\ \omega & \text{sonst.} \end{cases}$$

Die final produktive Fitness misst also den Fitnesswert, den ein Individuum über seine Nachkommen in die letzte Generation des Optimierungsprozesses einbringen kann. Wir argumentieren formal [GL20] und zeigen empirisch [GPL21], dass die final produktive Fitness die ideale Fitnessfunktion für einen evolutionären Algorithmus darstellen könnte. Freilich ist sie ohne hellseherische Fähigkeiten nicht praktisch zur Optimierung einsetzbar, da die Mengen D_x und X_n beide erst nach Abschluss der Evolution zur Verfügung stehen.



(a) Beste Werte der Zielfunktion t.



(b) Durschnittlicher Unterschied zwischen verwendeter Fitness f und faktischer final produktiver Fitness $\phi^\dagger$ pro Individuum.

Abb. 3: Evolutionsläufe für das klassische Problem von Schwefel. Eine naive Evolution mit $f(x, \_) = t(x)$ in Schwarz; diversitätsbasierte Evolution mit $f(x, X) = 0.5 \cdot t(x) + 0.5 \cdot d(x, X)$ in Blau; die Verfahren *inherited fitness* und *fitness sharing* in respektive lila und orange. Es wurden jeweils 20 Durchläufe ausgeführt; ausgefüllte Linien zeigen den Durchschnitt, transparente Linien das Band einer Standardabweichung. Bild aus [GPL21].

Doch wenigstens können wir diese Größe *a posteriori* abschätzen[5]. Wie Abbildung 3 zeigt, ist diejenige Fitnessfunktion für das beste Ergebnis verantwortlich, die über die Evolution hinweg die final produktive Fitness am *stabilsten* approximiert. Wir können deswegen vermuten, dass die final produktive Fitness am besten *a priori* abzuschätzen der ideale Nutzen jeder adaptierten Fitnessfunktion ist [GL20; GPL21].

4 Koevolutionäre Adaption von Fitness

Im vorangehenden Abschnitt konnten wir ein besseres Wissen um intrinsische Ziele zwar für eine bessere Performanz bei der Optimierung nutzen, mussten als Designer der Algorithmen jedoch auch komplexere Fitnessfunktionen definieren. In diesem Abschnitt zeigen wir, dass wir die exakte Fitnessfunktion auch durch eine eigene parallele Evolution steuern können. Wir nennen diesen Aufbau einen *koevolutionären Prozess*. Abbildung 4 zeigt ein Beispiel für so einen Prozess, das wir *Szenarien-Koevolution* (*scenario co-evolution*) nennen. Die grundlegende Optimierung findet in einem Prozess des verstärkenden Lernens statt: Ein virtueller Roboter soll eine virtuelle Fabrik durchlaufen, um bestimmte Ziele aufzusuchen. Auch dieser Prozess kann als Spezialfall eines evolutionären Prozesses betrachtet werden. In der virtuellen Fabrik können an jedem Ort zufällig Hindernisse erscheinen, die den Roboter beim Erreichen seiner Ziele behindern. Eine naive Simulation würde also zufällige Hindernisse mitsimulieren und den Roboter so im Laufe der Zeit auf deren Auftreten und seine richtige Reaktion trainieren. Wir haben gezeigt, dass die Lernergebnisse des Roboters jedoch besser ausfallen, wenn er nicht gegen zufällige sondern gegen möglichst schwierige Hindernisse trainiert [Ga19b]; leichte Konfigurationen von Hindernissen löst er dann ohnehin. Um möglichst schwere Konfigurationen für Hindernisse zu finden, nutzen wir einen evolutionären Algorithmus, der parallel zu dem verstärkenden Lernen läuft. Wir sprechen von einer *kompetitiven Koevolution*, weil Hinderniskonfigurationen danach bewertet werden, wie schlecht sie den aktuell besten Roboter werden lassen, und der Roboter danach bewertet wird, wie gut er mit den aktuell schwierigsten Hindernissen zurecht kommt. Damit sind die Fitness für Roboter und die Fitness für Hinderniskonfigurationen jeweils voneinander abhängig und das System aus beiden weist eine selbst-adaptive Fitness auf.

Obwohl der Mehraufwand an Rechenzeit durch den evolutionären Algorithmus durchaus erheblich ist, ist der bessere Lernfortschritt durch Szenarienkoevolution doch den Aufwand wert. Abbildung 5 zeigt, dass Szenarienkoevolution tendenziell bessere Ergebnisse pro Lernzeit erreicht.

Während wir das Schema eines koevolutionären Prozesses hier auf einen konkreten verstärkenden Lerner und einen evolutionären Algorithmus anwenden, könnte es sich auch

[5] Wir klammern hier eine kleine Diskussion aus: Final produktive Fitness schätzt den Wert eines Individuums über all seine möglichen Nachkommen in allen möglichen Folgeevolutionen ab, was natürlich noch viel mehr Berechnungskomplexität mit sich bringt. Mit der *faktischen* final produktiven Fitness approximieren wir diesen Wert nur noch, in dem wir ihn für nur eine faktisch beobachtete Folgeevolution berechnen und schlicht annehmen, dass diese ausreichend repräsentativ war.

als generelles Architekturmuster zur Entwicklung und dem Test von adaptiven System eignen [Ga20]. Insbesondere ist auch eine Diskussion zu verschiedenen Formen von adaptivem Verhalten und dessen Absicherung (idealerweise koevolutionär durch weitere adaptive Komponenten) Teil der hier beschriebenen Doktorarbeit [Ga16; Ga18; Ga20].

5 Schluss

Wir haben gezeigt, dass selbstadaptive Fitness natürlicherweise entsteht, dass sie hilft eine optimale Fitnessfunktion (für eine gegebene Zielfunktion) anzunähern, und dass wir angepasste Fitnessfunktionen effizient dynamisch adaptieren können, sogar für grundsätzlich verschiedene Formen von Lernen und Evolution. Die zugrundeliegende Doktorarbeit geht noch auf weitere Einsatzmöglichkeiten für (selbst-)adaptive Zielfunktionen ein und bietet ein deutlich solideres formales Fundament für unsere Definitionen.

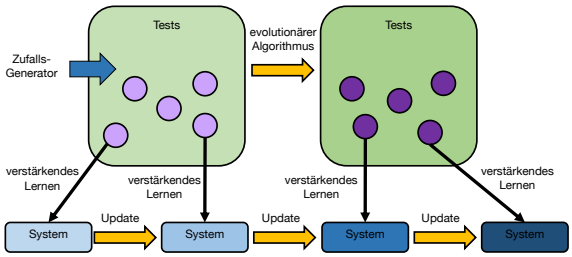


Abb. 4: Schematische Darstellung von Szenarien-Koevolution. Eine Population aus Testszenarien wird zunächst zufällig erzeugt und dann durch einen evolutionären Algorithmus stetig verbessert, während sich ein Agent mit verstärkendem Lernen auf zunehmend schwierigere Testszenarien einstellen kann. Bild aus [Ga19b].

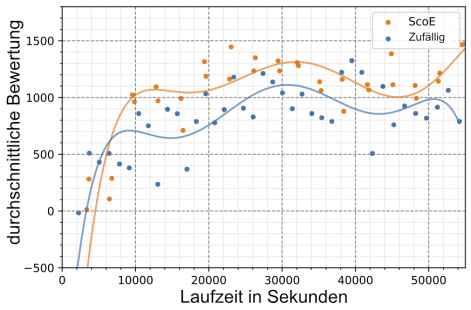


Abb. 5: Bewertungen, die von Szenarienkoevolution und einem naiven verstärkendem Lerner (der gegen zufällige Testzenarien trainiert) in der Fabrik-Hindernis-Testdomäne erreicht wurden, wobei beide gegen zufällige Testzenarien evaluiert wurden. Die Grafik zeigt einzelne Durchläufe und eine Trendkurve. Bild aus [Ga19b].

Aktuelle Arbeiten zielen nun darauf ab, selbst-replizierende Netze für nicht-triviale Lernaufgaben einzusetzen und ihre besonderen Eigenschaften und möglichen Vorteile genau zu analysieren. Eine systematische Betrachtung des Zusammenhangs zwischen extrinischer Zielfunktion und (teilweise) intrinsischer Fitness scheint für viele Anwendungsgebiete lange überfällig und eine Übertragbarkeit der final produktiven Fitness auf andere Techniken des maschinellen Lernens jenseits evolutionärer Algorithmen bleibt zu untersuchen. Koevolutionäre Ansätze finden sich mittlerweile in vielen Algorithmen im Bereich der künstlichen Intelligenz. Ein generelles Framework, das in der Lage ist hierüber viele verschiedene Lernalgorithmen sinnvoll zu kombinieren, bleibt eine Aufgabe für zukünftige Arbeiten.

Literatur

[BY15] Banzhaf, W.; Yamamoto, L.: Artificial Chemistries. MIT Press, 2015.

[CL18] Chang, O.; Lipson, H.: Neural Network Quine. In: ALIFE 2018: The 2018 Conference on Artificial Life. MIT Press, S. 234–241, 2018.

[Da09] Darwin, C.: The Origin of Species. PF Collier & son New York, 1909.

[Da96] Dawkins, R.: The Blind Watchmaker: Why the Evidence of Evolution Reveals a Universe Without Design. WW Norton & Company, 1996.

[De17] Dennett, D. C.: From Bacteria to Bach and Back: The Evolution of Minds. WW Norton & Company, 2017.

[DZB01] Dittrich, P.; Ziegler, J.; Banzhaf, W.: Artificial Chemistries—A Review. Artificial life 7/3, S. 225–275, 2001.

[FB96] Fontana, W.; Buss, L. W.: The Barrier of Objects: From Dynamical Systems to Bounded Organizations, 1996.

[Ga16] Gabor, T.; Belzner, L.; Kiermeier, M.; Beck, M. T.; Neitz, A.: A Simulation-Based Architecture for Smart Cyber-Physical Systems. In: The International Workshop on Models@run.time for Self-Aware Computing Systems. 2016.

[Ga18] Gabor, T.; Belzner, L.; Phan, T.; Schmid, K.: Preparing for the Unexpected: Diversity Improves Planning Resilience in Evolutionary Algorithms. In: 15th IEEE International Conference on Autonomic Computing (ICAC). 2018.

[Ga19a] Gabor, T.; Illium, S.; Mattausch, A.; Belzner, L.; Linnhoff-Popien, C.: Self-Replication in Neural Networks. In: Artificial Life Conference Proceedings. MIT Press, S. 424–431, 2019.

[Ga19b] Gabor, T.; Sedlmeier, A.; Kiermeier, M.; Phan, T.; Henrich, M.; Pichlmair, M.; Kempter, B.; Klein, C.; Sauer, H.; Schmid, R.; Wieghardt, J.: Scenario Co-Evolution for Reinforcement Learning on a Grid World Smart Factory Domain. In: Proceedings of the Genetic and Evolutionary Computation Conference. S. 898–906, 2019.

[Ga20] Gabor, T.; Sedlmeier, A.; Phan, T.; Ritz, F.; Kiermeier, M.; Belzner, L.; Kempter, B.; Klein, C.; Sauer, H.; Schmid, R.; Zeller, M.; Linnhoff-Popien, C.: The Scenario Coevolution Paradigm: Adaptive Quality Assurance for Adaptive Systems. International Journal on Software Tools for Technology Transfer/, S. 1–20, 2020.

[Ga21a] Gabor, T.: Self-Adaptive Fitness in Evolutionary Processes, Dissertation, Ludwig-Maximilians-Universität München, 2021.

[Ga21b] Gabor, T.; Illium, S.; Zorn, M.; Linnhoff-Popien, C.: Goals for Self-Replicating Neural Networks. In: ALIFE 2021: The 2021 Conference on Artificial Life. MIT Press, 2021.

[GB17] Gabor, T.; Belzner, L.: Genealogical Distance as a Diversity Estimate in Evolutionary Algorithms. In: Measuring and Promoting Diversity in Evolutionary Algorithms (MPDEA@GECCO). ACM, 2017.

[GBL18] Gabor, T.; Belzner, L.; Linnhoff-Popien, C.: Inheritance-Based Diversity Measures for Explicit Convergence Control in Evolutionary Algorithms. In: The Genetic and Evolutionary Computation Conference (GECCO). 2018.

[GL20] Gabor, T.; Linnhoff-Popien, C.: A Formal Model for Reasoning about the Ideal Fitness in Evolutionary Processes. In: International Symposium on Leveraging Applications of Formal Methods (ISoLA). 2020.

[GPL21] Gabor, T.; Phan, T.; Linnhoff-Popien, C.: Productive Fitness in Diversity-Aware Evolutionary Algorithms, 2021.

[Kr11] Kruse, R.; Borgelt, C.; Klawonn, F.; Moewes, C.; Ruß, G.; Steinbrecher, M.; Held, P.: Computational Intelligence. Springer, 2011.

[WO03] Wineberg, M.; Oppacher, F.: The Underlying Similarity of Diversity Measures Used in Evolutionary Computation. In: Genetic and Evolutionary Computation Conference. Springer, S. 1493–1504, 2003.

Thomas Gabor studierte von 2009 bis 2015 Informatik an der Ludwig-Maximilians-Universität München. Von 2015 bis 2021 promovierte er ebenda bei Claudia Linnhoff-Popien und unterstützte den Aufbau der Themenfelder *Artificial Intelligence* und *Quantum Computing* am Lehrstuhl für mobile und verteilte Systeme. Dabei betreute er auch zahlreiche Industrie- und Förderprojekte. Er ist Mitgründer der 2021 am Lehrstuhl entstandenen Ausgründung Aqarios, die Softwarewerkzeuge (u.A. Optimierungsalgorithmen) für das Zeitalter der Quantencomputer entwickelt. Für seine Disseration wurde er 2021 mit dem Heinz-Schwärtzel-Preis ausgezeichnet. Als Postdoc bereitet er für 2022 eine Vorlesung zu *Natural Computing* vor.

Stern-Topologie-Entkoppelte Zustandsraumsuche in der KI-Planung und Modellprüfung[1]

Daniel Gnad[2]

Abstract: Die Zustandsraumsuche ist ein weit verbreitetes Konzept in vielen Bereichen der Informatik. Die Größe der zu durchsuchenden Zustandsräume wächst jedoch typischerweise exponentiell mit der Größe einer kompakten, faktorisierten Modellbeschreibung – das ist das bekannte Problem der Zustandsexplosion. Die Entkoppelte Zustandsraumsuche (entkoppelte Suche) beschreibt einen neuartigen Ansatz um der Zustandsexplosion entgegenzuwirken. Hierfür wird die Struktur des Modells, insbesondere die bedingte Unabhängigkeit von Systemkomponenten in einer Sterntopologie, ausgenutzt. Diese Unabhängigkeit ergibt sich ganz natürlich bei vielen faktorisierten Modellen deren Zustandsräume aus dem Produkt mehrerer Komponenten bestehen. In der Dissertation wird die entkoppelte Suche in der Planung – als Teil der Künstlichen Intelligenz (KI) – und in der Verifikation mittels Modellprüfung eingeführt. Das Konzept des entkoppelten Zustandsraums wird auf Basis von etablierten Formalismen entwickelt und seine Korrektheit bezüglich der exakten Erfassung der Erreichbarkeit von Modellzuständen bewiesen. Damit kann die entkoppelte Suche mit beliebigen Suchalgorithmen genutzt und mit komplementären Techniken kombiniert werden. In der Dissertation wird gezeigt dass die entkoppelte Suche den Suchaufwand exponentiell stärker reduzieren kann als existierende alternative Ansätze, insbesondere die Reduktion partieller Ordnung, Symmetriereduktion, Entfaltung von Petri-Netzen und symbolische Suche. Empirisch kann die entkoppelte Suche sowohl in der Planung als auch in der Modellprüfung etablierte Systeme deutlich übertreffen.

1 Einführung

Eine Vielzahl von Problemen in der Informatik kann als Suche in einem Zustandsraum formuliert werden. Die Zustandsräume dienen dann als formale Spezifikation des möglichen Verhaltens der betrachteten Systeme und werden typischerweise kompakt als Menge von miteinander interagierenden Komponenten modelliert. Da die Größe der Zustandsräume im Allgemeinen jedoch exponentiell mit der Größe der Spezifikation wächst – das Problem der Zustandsexplosion – müssen Zustandsräume möglichst effizient systematisch exploriert werden um – beispielsweise in der Planung – eine Sequenz von Aktionen zu finden, die zu einem Zustand mit gewissen Zieleigenschaften führt oder um – wie in der Modellprüfung – gewünschte Eigenschaften des Systems zu verifizieren.

In der Dissertation [Gn21] wird eine neuartige Methode einführt, die der Zustandsexplosion entgegen wirkt, die *Stern-Topologie-entkoppelte Zustandsraumsuche*, kurz entkoppelte Suche (engl. decoupled search). Es werden zwei etablierte Formalismen betrachtet,

[1] Englischer Titel: Star-Topology Decoupled State-Space Search in AI Planning and Model Checking

[2] Universität von Linköping, Schweden, daniel.gnad@liu.se

um Zustandsräume zu beschreiben. Hauptfokus der Arbeit liegt auf der Handlungsplanung [GNT04], in der Zustände als Zuweisung an eine Menge von Variablen definiert sind, deren Werte durch Aktionen geändert werden können. Außerdem wird die entkoppelte Suche im Kontext der Modellprüfung [CGP01] eingeführt, in der Systeme als Menge synchronisierter nicht-deterministischer Automaten modelliert sind.

Die Grundidee der entkoppelten Suche besteht darin, das faktorisierte Modell, also die Zustandsvariablen im Planen, bzw. die Automaten in der Modellprüfung, so zu dekomponieren, dass die Interaktionen zwischen den Komponenten die Form einer Sterntopologie, mit einer zentralen Komponente C und einer beliebigen Anzahl Blattkomponenten $\mathcal{L} = \{L_1, \ldots, L_n\}$, annehmen. In dieser Topologie dürfen Blätter beliebig mit dem Zentrum interagieren, jede direkte Interaktion zwischen Blättern muss jedoch auch das Zentrum einschließen. Die wesentliche Beobachtung ist, dass die Blattkomponenten in dieser Topologie *bedingt unabhängig* voneinander sind. Dadurch kann die Suche auf Transitionen beschränkt werden, die das Zentrum betreffen und entlang einer Sequenz solcher Transitionen können – für jedes Blatt $L \in \mathcal{L}$ separat – alle *konformen* Sequenzen von L-Transitionen, also Transitionen die L ändern, aufgezählt werden. Suchknoten entsprechen dann sogenannten *entkoppelten Zuständen*, welche eine potentiell exponentiell große Menge expliziter Zustände des Modells repräsentieren. Dadurch lässt sich der Suchaufwand signifikant reduzieren.

Im Folgenden schauen wir uns die entkoppelte Suche in der Handlungsplanung sowie der Modellprüfung an. Dies entspricht dem ersten und dritten Teil der Dissertation. Anschließend geben wir einen kurzen Überblick zum zweiten Teil der Arbeit, der sich mit der Kombination der entkoppelten Suche mit etablierten Konkurrenzmethoden im Kontext der Planung befasst, der partiellen Ordnungsreduktion [GW91; Va89], der Symmetriereduktion [ES96; St91], der symbolischen Suche mittels binärer Entscheidungsdiagramme (BDDs) [Br86; Mc93], sowie der Dominanzreduktion mit Simulationsrelationen [Mi71].

2 Entkoppelte Suche in der Handlungsplanung

Die Handlungsplanung, kurz Planung, beschäftigt sich mit dem Finden einer Sequenz von Aktionen die ausgehend von einem Startzustand eine gewünschte Zielbedingung erreichen. Konkret wird hier die *klassische* Handlungsplanung im FDR Formalismus betrachtet [BN95; He06], in dem Aktionen diskret sind und deterministische Effekte haben, die vollständig beobachtbar sind. Einer der besten Ansätze zum Lösung von Planungsaufgaben ist die heuristische Suche, in welcher beginnend in einem Startzustand $\mathcal{I}$, der Zustandsraum der Aufgabe mittels Heuristik systematisch durchsucht wird bis ein Zustand s erreicht wurde, der die Zielbedingung $\mathcal{G}$ erfüllt, falls ein solcher Zustand von $\mathcal{I}$ erreichbar ist. Die Lösung der Planungsaufgabe, ein *Plan*, ist die Aktionssequenz mit der s erreicht wurde.

Definition (Planungsaufgabe)**.** Eine *Planungsaufgabe* ist ein Tupel $\Pi = \langle \mathcal{V}, \mathcal{A}, \mathsf{cost}, \mathcal{I}, \mathcal{G} \rangle$, wobei $\mathcal{V}$ eine endliche Menge von *Zustandsvariablen* ist, jede Variable $v \in \mathcal{V}$ hat eine endliche *Domäne* $\mathcal{D}(v)$. Eine vollständige Zuweisung an $\mathcal{V}$ wird *Zustand* genannt, eine partielle

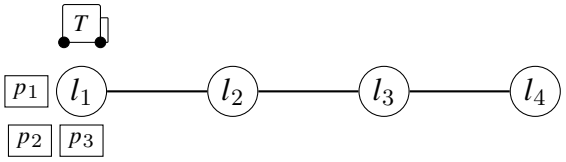


Abb. 1: Eine grafische Illustration des initialen Zustands unseres Beispiels.

Zuweisung p an eine Teilmenge $\mathsf{vars}(p) \subseteq \mathcal{V}$ der Variablen ist ein *partieller Zustand*. $\mathcal{A}$ ist eine endliche Menge von *Aktionen*. Jede Aktion $a \in \mathcal{A}$ ist ein Paar $\langle \mathsf{pre}(a), \mathsf{eff}(a) \rangle$, wobei $\mathsf{pre}(a)$ die *Vorbedingung* (engl. precondition) von a ist, und $\mathsf{eff}(a)$ der *Effekt*, beides sind partielle Zustände. Die *Kostenfunktion* $\mathsf{cost} : \mathcal{A} \to \mathbb{R}^{0+}$ weist jeder Aktion $a \in \mathcal{A}$ ihre nicht-negativen reellen Kosten $\mathsf{cost}(a)$ zu. $\mathcal{I}$ ist der *Initiale Zustand* und $\mathcal{G}$ die *Zielbedingung* (engl. goal), ein partieller Zustand.

Die Zustandsvariablen $\mathcal{V}$ beschreiben die Menge der Zustände von Π, Aktionen beschreiben die Übergänge zwischen Zuständen. Dadurch ergibt sich der Zustandsraum von Π, dessen Größe exponentiell mit der Anzahl Variablen wächst. Das Planexistenzproblem, also das Entscheidungsproblem, ob ein Plan für eine Aufgabe existiert, ist **PSPACE**-vollständig.

Zur Illustration schauen wir uns ein Logistikbeispiel an, bei dem ein LKW T drei Pakete, p_1, p_2, p_3, zu deren Zielposition liefern soll. Abbildung 1 zeigt grafisch den initialen Zustand des Beispiels. Formal kann das Beispiel wie folgt definiert werden: die Zustandsvariablen sind $\mathcal{V} = \{T, p_1, p_2, p_3\}$, initialer Zustand $\mathcal{I} = \{T = l_1, p_1 = l_1, p_2 = l_1, p_3 = l_1\}$, Zielbedingung $\mathcal{G} = \{p_1 = l_4, p_2 = l_4, p_3 = l_4\}$ und Aktionen (alle mit Kosten 1).

$$\begin{aligned}\mathcal{A} = \{&\mathsf{drive}(a,b) \mid \{a,b\} \in \{\{l_1,l_2\},\{l_2,l_3\},\{l_3,l_4\}\}\} \cup \\ &\{\mathsf{load}(p_j,l_k), \mathsf{unload}(p_j,l_k) \mid j \in \{1,2,3\}, k \in \{1,2,3,4\}\}, \text{wobei:} \\ &\quad \mathsf{pre}(\mathsf{drive}(a,b)) = \{T = a\}, \mathsf{eff}(\mathsf{drive}(a,b)) = \{T = b\}, \\ &\quad \mathsf{pre}(\mathsf{load}(p_j,l_k)) = \{T = l_k, p_j = l_k\}, \mathsf{eff}(\mathsf{load}(p_j,l_k)) = \{p_j = T\}, \\ &\quad \mathsf{pre}(\mathsf{unload}(p_j,l_k)) = \{T = l_k, p_j = T\}, \mathsf{eff}(\mathsf{unload}(p_j,l_k)) = \{p_j = l_k\}.\end{aligned}$$

Ein möglicher Plan für diese Aufgabe lädt die drei Pakete in den LKW, fährt diesen nach l_4 und lädt die Pakete dort aus. Hier ergeben sich allein für das Laden der Pakete in l_1 sechs Möglichkeiten, die jedoch keinen Einfluss auf die Existenz bzw. die Kosten eines Plans haben. Trotzdem muss die Suche all diese Möglichkeiten betrachten. Skaliert man die Anzahl Pakete, so ergibt sich eine exponentielle Zahl an Zuständen die sich nur darin unterscheiden, in welcher Reihenfolge die Pakete in l_1 geladen werden. Genau dieses Problem geht die entkoppelte Suche an. Hierfür werden zunächst die Abhängigkeiten zwischen den Zustandsvariablen untersucht um diese zu partitionieren. Damit kann die nötige Struktur für die entkoppelte Suche, eine Stern-Faktorisierung, identifiziert werden:

Definition (Stern-Faktorisierung)**.** Sei Π eine Planungsaufgabe. Eine *Stern-Faktorisierung* ist eine Partitionierung der Variablen $\mathcal{V}$ in Faktoren $\mathcal{F}$, so dass folgende Eigenschaften

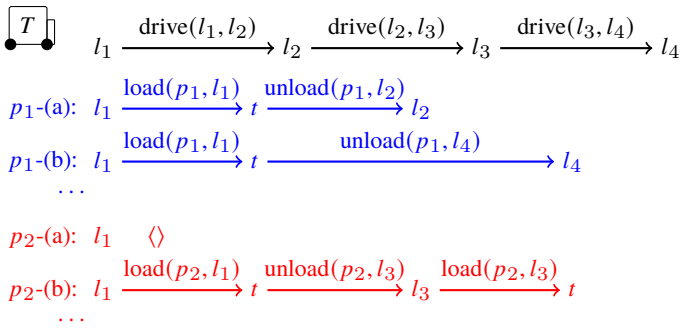


Abb. 2: Illustration einer Teilmenge der konformen *leaf*-Pfade entlang eines *center* Pfades.

gelten: es existiert ein *center* Faktor $C \in \mathcal{F}$, der Rest der Faktoren $\mathcal{L} := \mathcal{F} \setminus \{C\}$ wird als *leaves* bezeichnet, und für alle Aktionen $a \in \mathcal{A}$ gilt: entweder $\mathsf{vars}(\mathsf{eff}(a)) \cap C \neq \emptyset$ oder es existiert ein *leaf* $L \in \mathcal{L}$ so dass $\mathsf{vars}(\mathsf{eff}(a)) \subseteq L$ und $\mathsf{vars}(\mathsf{pre}(a)) \cup \mathsf{vars}(\mathsf{eff}(a)) \subseteq C \cup L$.

In einer Stern-Faktorisierung dürfen Aktionen beliebige Faktoren betreffen, so lange sie auch den *center* Faktor ändern (*center*-Aktionen). Die restlichen Aktionen (*leaf*-Aktionen) dürfen nur ein einziges *leaf* L ändern und nur Vorbedingungen auf L sowie C haben.

Die wesentliche Beobachtung für die entkoppelte Suche ist dass die *leaf* Faktoren bedingt unabhängig voneinander sind. Dies erlaubt es die Suche auf *center*-Aktionen zu beschränken und die möglichen *konformen leaf*-Pfade entlang einer Sequenz von *center*-Aktionen für jedes *leaf* separat aufzuzählen. Hierbei wir ein *leaf*-Pfad π^L als *konform* mit einem *center*-Pfad π^C bezeichnet, wenn die Teilsequenzen geteilter Aktionen übereinstimmt[3] und π^L in π^C eingebettet werden kann, so dass die resultierende Aktionssequenz einen korrekten Pfad in der Projektion von Π auf $C \cup L$ darstellt. Suchknoten entsprechen dann sogenannten *entkoppelten Zuständen*, den Endpunkten von *center*-Pfaden. Ein entkoppelter Zustand $s^{\mathcal{D}}$ ist definiert als Tupel $\langle \pi^C, s^C, \mathsf{prices}(s^{\mathcal{D}}) \rangle$, bestehend aus dem *center*-Pfad auf dem $s^{\mathcal{D}}$ erreicht wurde, einem *center*-Zustand, einer Zuweisung an C, sowie der *pricing* Funktion, die jedem *leaf*-Zustand s^L, einer Zuweisung an ein $L \in \mathcal{L}$, die minimalen Kosten eines konformen *leaf*-Pfades π^L zuweist auf dem s^L entlang von π^C erreicht werden kann.

Eine mögliche Stern-Faktorisierung unseres Logistikbeispiels besteht aus dem *center* $C = \{T\}$ und drei *leaf* Faktoren $L_i = \{p_i\}$, einem für jedes Paket. Für diese Faktorisierung zeigt Abbildung 2 eine (kleine) Teilmenge der konformen *leaf*-Pfade von L_1 und L_2 für den *center*-Pfad, der den LKW von l_1 nach l_4 fährt. Die Hauptbeobachtung ist hier, dass unabhängig davon, welchen der beiden Pfade – (a) oder (b) – wir für Paket p_1 wählen, ein beliebiger konformer Pfad für Paket p_2 gewählt werden kann, da die Faktoren unabhängig voneinander sind. Die konformen *leaf*-Pfade werden mittels der *pricing* Funktion kompakt repräsentiert. So hat beispielsweise der *leaf*-Zustand $(p_1 = l_3)$ im Endzustand des in

[3] In komplexeren Topologien haben *center*-Aktionen oft *leaf*-Effekte, und zählen somit auch zu den *leaf*-Aktionen.

			Blinde Suche							A^* mit $h^{\text{LM-cut}}$							
				Explizite Suche						Explizite Suche							
Domäne	#	$\#\mathcal{F}$	unf	b	pp	por	sym	p+s	DS	b	pp	por	sym	p+s	DS	sbd	c2
Driverlog	20	20	4	7	7	7	7	7	**11**	13	14	13	13	13	13	12	**15**
Logistics	63	63	11	12	14	12	14	14	**26**	26	26	27	26	28	**33**	24	28
Miconic	150	145	25	45	45	40	**51**	43	46	136	136	136	**137**	136	135	107	98
NoMystery	20	20	4	8	8	8	8	8	**20**	14	14	14	15	14	**20**	14	**20**
Pathways	30	29	2	**3**	**3**	**3**	**3**	**3**	**3**	**4**	**4**	**4**	**4**	**4**	**4**	**4**	**4**
Rovers	40	40	2	5	6	**7**	5	**7**	6	7	10	9	7	9	9	**14**	13
Satellite	36	36	**7**	5	**7**	6	6	6	6	7	12	11	13	**14**	8	8	9
TPP	30	27	3	5	5	5	6	6	**23**	5	6	5	7	6	**23**	7	14
Woodwork	30	13	**7**	4	4	6	4	**7**	5	6	6	**11**	7	**11**	**11**	9	9
Zenotravel	20	20	7	8	8	7	8	7	**12**	**13**	**13**	**13**	**13**	**13**	12	10	**13**
Andere	1191	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
Summe	1630	417	76	106	111	105	116	112	**162**	235	245	247	246	252	**272**	213	227

Tab. 1: Anzahl gelöster Planungsinstanzen im optimalen Planen. $\#\mathcal{F}$ ist die Anzahl Instanzen mit *fork* Faktorisierung. Verglichen wird die entkoppelte Suche (DS) mit verschiedenen anderen Ansätzen im Kontext der blinden Suche (links), sowie der heuristischen Suche mit $h^{\text{LM-cut}}$ (rechts).

Abb. 2 gezeigten *center* Pfades den Preis 2, entsprechend der Kosten des konformen Pfades $\langle \text{load}(p_1, l_1), \text{unload}(p_1, l_3) \rangle$. Beide Pakete können auch in ihrer initialen Position mit Preis 0 verbleiben, da hierfür keine Aktion angewendet werden muss.

Mit der genannten Faktorisierung besteht der gesamte *entkoppelten Zustandsraum* unseres Beispiels aus nur zehn entkoppelten Zustände, im Gegensatz zu 500 expliziten Zuständen. Skaliert man das Modell auf N Pakete hoch, so hat dies *keinen* Einfluss auf die Größe des entkoppelten Zustandsraums, wohingegen der explizite Zustandsraum exponentiell in N wächst. In diesem, zugegebenermaßen simplen, Beispiel führt die entkoppelte Suche also zu einer exponentiellen Reduktion des Zustandsraums.

2.1 Eigenschaften der Entkoppelten Suche & verwandte Arbeiten

Die entkoppelte Suche kann mit einem beliebigen Suchalgorithmus genutzt werden und erlaubt die Nutzung existierender Planungsheuristiken. Die Suche im entkoppeltem Zustandsraum erhält hierbei alle Eigenschaften der Suchalgorithmen und Heuristiken, wie Korrektheit, Vollständigkeit und Optimalität. Die entkoppelte Suche ist also eine exakte Methode, welche die Erreichbarkeit (und Kosten) aller Zustände erhält.

Satz 1. Die entkoppelte Suche erhält die Korrektheit, Vollständigkeit und Optimalität des genutzten Suchalgorithmus sowie die Sicherheit und Zulässigkeit von Planungsheuristiken.

Im Vergleich zu verwandten Reduktionsmethoden kann die entkoppelte Suche zu einer exponentiell stärkeren Reduktion führen. In der Dissertation wird dies anhand von skalierba-

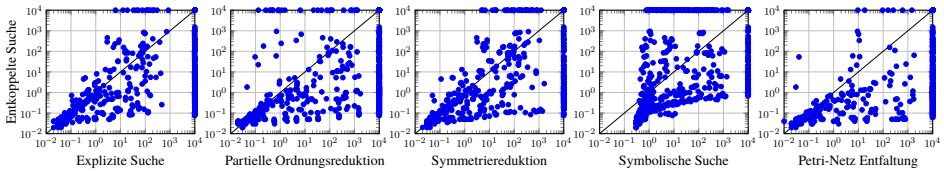


Abb. 3: Vergleich der Laufzeit (in Sekunden) zum kompletten Aufbau des Zustandsraums von entkoppelter Suche (DS) auf der y-Achse und verschiedenen alternativen Methoden auf der x-Achse.

ren Planungsaufgaben formalisiert, so dass der entkoppelte Zustandsraum nur polynomiell mit der Größe des Modell wächst, der Zustandsraum der jeweils anderen Methode jedoch exponentiell. Ist dies der Fall, so nennen wir die beiden Methoden *exponentiell separiert*.

Satz 2. Die entkoppelte Suche ist exponentiell separiert von der partiellen Ordnungsreduktion mittels *strong stubborn sets*, der Symmetriereduktion, der symbolischen Suche mit BDDs, der Dominanzreduktion sowie der Entfaltung von Petri-Netzen, und umgekehrt.

2.2 Empirische Auswertung

Die entkoppelte Suche ist im Fast Downward System [He06] integriert, die Implementierung ist frei verfügbar (`https://gitlab.com/dgnad/decoupled-fast-downward/`).

Tabelle 1 zeigt Resultate im optimalen Planen, wo Pläne mit minimaler Summe an Aktionskosten zurückgegeben werden müssen. Wir zeigen Ergebnisse für zwei Suchstrategien, blinde Suche ohne Heuristik sowie A^*-Suche mit der $h^{\text{LM-cut}}$ Heuristik [HD09]. Verglichen wird die entkoppelte Suche (**DS**), mit der Petri-Netz Entfaltung (**unf**), expliziter Suche ohne Reduktionstechnik (**b**), bzw. mit partieller Ordnungsreduktion (**por**), Partitionsreduktion – einer verwandten Methode aus der Handlungsplanung – (**pp**), Symmetriereduktion (**sym**) sowieso einer Kombination von por und sym (**p+s**). Außerdem schließen wir die bidirektionale symbolische Suche (**sbd**) und den Complementary2 Planer [FLB18] (**c2**) in den Vergleich ein. Die Tabelle zeigt, pro Domäne, die Anzahl gelöster Instanzen.

Abbildung 3 zeigt einen Laufzeitvergleich von entkoppelter Suche mit den Alternativmethoden. Szenario ist der komplette Aufbau des erreichbaren Zustandsraums. Jeder Punkt im Diagramm entspricht einer Probleminstanz, Punkte unterhalb der Diagonalen zeigen Instanzen, in denen die entkoppelte Suche schneller ist, als die Konkurrenzmethode.

Sowohl Tabelle 1 als auch Abbildung 3 zeigen klar, dass die entkoppelte Suche alle gezeigten Methoden deutlich übertreffen kann. Die Laufzeitdiagramme sind im logarithmischen Maßstab abgebildet, es ergibt sich also oft ein Vorteil von entkoppelter Suche um mehrere Größenordnungen im Vergleich zu allen gezeigten Methoden.

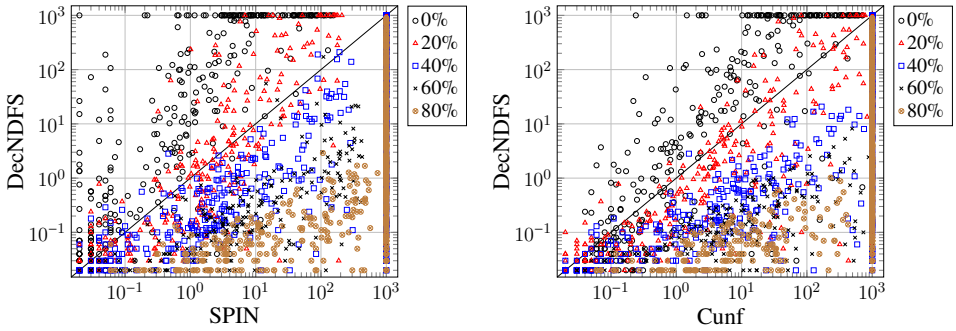


Abb. 4: Laufzeitstatistik, je Probleminstanz, für die Prüfung von Lebensdigkeitseigenschaften von entkoppelter Suche, DecNDFS, auf der y-Achse, gegen SPIN (links) und Cunf (rechts), jeweils auf der x-Achse. Farblich hervorgehoben ist der Anteil an internen Transitionslabels.

3 Entkoppelte Suche in der Modellprüfung

Wie in der Planung steht auch in der Modellprüfung die Frage nach der Dekomposition des Modells an erster Stelle. Betrachtet werden Systeme mit mehreren nicht-deterministischen Automaten, die auf einer Teilmenge ihrer Transitionslabels synchronisieren. Hierbei müssen geteilte Transitionen immer synchron ausgeführt werden, Transitionen die nur einen Automaten betreffen können unabhängig von diesem genommen werden. Eine Faktorisierung der Modelle entsteht durch die Betrachtung jedes Automaten als separate Komponente, woraus sich ganz natürlich eine Aufteilung in globale, synchronisierende, und interne Transitionen ergibt (in der Planung entsprechend *center*/*leaf* -Aktionen). Die entkoppelte Suche betrachtet dann nur globale Transitionen und zählt, separat für jeden Automaten, die Zustände auf, die über konforme interne Transitionssequenzen erreichbar sind.

Die Prüfung von Sicherheitseigenschaften, also der Erreichbarkeit von Zuständen mit gewissen Eigenschaften, entspricht algorithmisch der Erreichbarkeitprüfung von Zielzuständen in der Planung, es Bedarf keiner neuen Anpassungen des entkoppelten Zustandsraums. Zum Prüfen von Lebendigkeitseigenschaften, genauer ω-reguläre Eigenschaften, sind jedoch Anpassungen nötig. Die Dissertation entwickelt hierfür eine Anpassung des *nested depth-first search* (NDFS) Algorithmus [Co92] und beweist deren Korrektheit und Vollständigkeit. Außerdem wird gezeigt, dass die entkoppelte Suche auch zum Prüfen von Lebendigkeitseigenschaften exponentielle Vorteile gegenüber Konkurrenzmethoden wie der Reduktion partieller Ordnung sowie dem Entfalten von Petri-Netzen haben kann.

3.1 Empirische Auswertung

Im Kontext der Prüfung von Lebendigkeitseigenschaften vergleichen wir die entkoppelte Suche mit dem etablierten SPIN Modellprüfer [Ho04] und dem Cunf Petri-Netz Entfaltungstool [RS13]. Für letzteres schließt die Laufzeit lediglich den Aufbau eines vollständigen

Unfolding-Präfixes ein, nicht die Zeit zum Prüfen der Eigenschaft. Als Benchmark wird eine Menge zufällig generierter Automaten genutzt, wobei wir die Anzahl Komponenten, sowie den Anteil interner Transitionslabels skalieren. Abbildung 4 vergleicht die Laufzeit von entkoppelter Suche auf der y-Achse mit den beiden Alternativmethoden auf der x-Achse, wie schon im Abschnitt zur Planung. Es zeigt sich, dass ab einem Anteil interner Labels von ca. 20% die entkoppelte Suche konsistent schneller terminiert, um bis zu mehrere Größenordnungen mit hohem Anteil interner Labels. In der Dissertation wird ein ähnliches Bild gezeigt wenn statt der internen Labels die Anzahl der Modellkomponenten betrachtet wird. Ab vier Komponenten schlägt die entkoppelte Suche die anderen Methoden deutlich.

4 Kombination mit alternativen Methoden

Im zweiten Teil der Dissertation wird die entkoppelte Suche im Kontext der Handlungsplanung mit alternativen Methoden kombiniert. Da die entkoppelte Suche im Vergleich zu diesen Methoden eine andersartige Reduktion erzielt, erscheint es sinnvoll, zu untersuchen ob Synergien entstehen. Wie insbesondere anhand der Kombinationen mit der partiellen Ordnungsreduktion durch *strong stubborn sets* [Al12], der Symmetriereduktion via *orbit-space search* [DKS12; PZR11], der symbolischen Suche mit BDDs [JVB08; To17], sowie der Dominanzreduktion [TH15] zu sehen ist, ist dies der Fall.

Auf theoretischer Ebene konnte gezeigt werden, dass die kombinierten Algorithmen exponentielle Vorteile gegenüber ihren Komponenten erzielen können. Auch empirisch können die Kombinationen überzeugen, erben meist die Stärken ihrer besten Komponente und entwickeln positive Synergien, wo das Ganze mehr als die Summe seiner Teile ist.

5 Diskussion

In der Dissertation wird mit der entkoppelten Suche eine neuartige Methode zur Reduktion von Zustandsräumen entwickelt und im Kontext der Handlungsplanung sowie der Modellprüfung eingeführt. Die entkoppelte Suche ist dabei nicht auf ein Teilgebiet der Informatik beschränkt, sondern kann im Prinzip auf viele Arten von Problemen angewandt werden, nämlich solche, die als Suche in einem Zustandsraum formuliert werden können, der implizit als faktorisiertes Modell, also als Menge interagierender Komponenten, spezifiziert werden kann. Die entkoppelte Suche kann im Vergleich zu allen bekannten alternativen Methoden zu einer exponentiell stärkeren Reduktion führen. Dies gilt insbesondere für die Reduktion partieller Ordnung, Symmetriereduktion, symbolische Suche, Dominanzreduktion sowie die Entfaltung von Petri-Netzen. Es handelt sich also um einen neuartigen Ansatz, der die Modellstruktur auf eine neue Weise ausnutzt. Empirisch hat sich gezeigt, dass die entkoppelte Suche mit diesen Methoden, sowie generell mit dem aktuellen Stand der Technik in der Planung und Modellprüfung, nicht nur mithalten, sondern sie auf Probleminstanzen mit ausgeprägter Sterntopologie auch deutlich übertreffen kann. Die entkoppelte Suche stellt somit eine neue Option zur Analyse von sehr großen Zustandsräumen dar.

Literatur

[Al12] Alkhazraji, Y.; Wehrle, M.; Mattmüller, R.; Helmert, M.: A Stubborn Set Algorithm for Optimal Planning. In (Raedt, L. D., Hrsg.): Proceedings of the 20th European Conference on Artificial Intelligence (ECAI'12). IOS Press, Montpellier, France, S. 891–892, Aug. 2012.

[BN95] Bäckström, C.; Nebel, B.: Complexity Results for SAS^+ Planning. Computational Intelligence 11/4, S. 625–655, 1995.

[Br86] Bryant, R. E.: Graph-Based Algorithms for Boolean Function Manipulation. IEEE Transactions on Computers 35/8, S. 677–691, 1986.

[CGP01] Clarke, E.; Grumberg, O.; Peled, D.: Model Checking. MIT Press, 2001.

[Co92] Courcoubetis, C.; Vardi, M. Y.; Wolper, P.; Yannakakis, M.: Memory-Efficient Algorithms for the Verification of Temporal Properties. Formal Methods in System Design 1/2/3, S. 275–288, 1992.

[DKS12] Domshlak, C.; Katz, M.; Shleyfman, A.: Enhanced Symmetry Breaking in Cost-Optimal Planning as Forward Search. In (Bonet, B.; McCluskey, L.; Silva, J. R.; Williams, B., Hrsg.): Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS'12). AAAI Press, 2012.

[ES96] Emerson, E. A.; Sistla, A. P.: Symmetry and model-checking. Formal Methods in System Design 9/1/2, S. 105–131, 1996.

[FLB18] Franco, S.; Lelis, L. H.; Barley, M.: The Complementary2 Planner in the IPC 2018. In: IPC 2018 planner abstracts. 2018.

[Gn21] Gnad, D.: Star-topology Decoupled State-Space Search in AI Planning and Model Checking, Diss., Universität des Saarlandes, Saarbrücken, Germany, 2021.

[GNT04] Ghallab, M.; Nau, D.; Traverso, P.: Automated Planning: Theory and Practice. Morgan Kaufmann, 2004.

[GW91] Godefroid, P.; Wolper, P.: Using Partial Orders for the Efficient Verification of Deadlock Freedom and Safety Properties. In: Proceedings of the 3rd International Workshop on Computer Aided Verification (CAV'91). S. 332–342, 1991.

[HD09] Helmert, M.; Domshlak, C.: Landmarks, Critical Paths and Abstractions: What's the Difference Anyway? In (Gerevini, A.; Howe, A.; Cesta, A.; Refanidis, I., Hrsg.): Proceedings of the 19th International Conference on Automated Planning and Scheduling (ICAPS'09). AAAI Press, S. 162–169, 2009.

[He06] Helmert, M.: The Fast Downward Planning System. Journal of Artificial Intelligence Research 26/, S. 191–246, 2006.

[Ho04] Holzmann, G.: The Spin Model Checker - Primer and Reference Manual. Addison-Wesley, 2004.

[JVB08] Jensen, R. M.; Veloso, M. M.; Bryant, R. E.: State-set branching: Leveraging BDDs for heuristic search. Artificial Intelligence 172/2-3, S. 103–139, 2008.

[Mc93] McMillan, K. L.: Symbolic Model Checking. Kluwer Academic Publishers, 1993.

[Mi71] Milner, R.: An Algebraic Definition of Simulation Between Programs. In: Proceedings of the 2nd International Joint Conference on Artificial Intelligence (IJCAI'71). William Kaufmann, London, UK, S. 481–489, Sep. 1971.

[PZR11] Pochter, N.; Zohar, A.; Rosenschein, J. S.: Exploiting Problem Symmetries in State-Based Planners. In (Burgard, W.; Roth, D., Hrsg.): Proceedings of the 25th National Conference of the American Association for Artificial Intelligence (AAAI'11). AAAI Press, San Francisco, CA, USA, Juli 2011.

[RS13] Rodríguez, C.; Schwoon, S.: Cunf: A Tool for Unfolding and Verifying Petri Nets with Read Arcs. In: Proceedings of the 11th International Symposium on Automated Technology for Verification and Analysis (ATVA'13). S. 492–495, 2013.

[St91] Starke, P.: Reachability analysis of Petri nets using symmetries. Journal of Mathematical Modelling and Simulation in Systems Analysis 8/4/5, S. 293–304, 1991.

[TH15] Torralba, Á.; Hoffmann, J.: Simulation-Based Admissible Dominance Pruning. In (Yang, Q., Hrsg.): Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'15). AAAI Press/IJCAI, S. 1689–1695, 2015.

[To17] Torralba, Á.; Alcázar, V.; Kissmann, P.; Edelkamp, S.: Efficient symbolic search for cost-optimal planning. Artificial Intelligence 242/, S. 52–79, 2017.

[Va89] Valmari, A.: Stubborn sets for reduced state space generation. In: Proceedings of the 10th International Conference on Applications and Theory of Petri Nets. S. 491–515, 1989.

Daniel Gnad, geboren 1987, studierte Computer- und Kommunikationstechnik an der Universität des Saarlandes, wo er 2014 seinen Master erwarb. Anschließend schloss er sich dem Lehrstuhl der Grundlagen der Künstlichen Intelligenz (FAI) von Prof. Jörg Hoffmann als Promotionsstudent an und forschte im Bereich der klassischen KI Planung. Neben dem Thema seiner Dissertation, der entkoppelten Zustandsraumsuche, widmete er sich der Erforschung neuer Planungsheuristiken im Kontext der Red-Black Relaxierung, sowie dem Einsatz von Methoden des machinellen Lernens zur kompakteren Grundung von Planungsaufgaben. Im November 2021 schloss er seine Promotion mit *summa cum laude* ab. Seit Februar 2022 ist er Postdoc in der Gruppe für Künstliche Intelligenz und integrierte Computersysteme (AIICS) an der Universität von Linköping.

Programmaggregation mit algebraischen Entscheidungsdiagrammen[1]

Frederik Gossen[2]

Abstract: Im Rahmen dieser Dissertation wurde das Potential von algebraischen Entscheidungsdiagrammen (ADDs) als Programmrepräsentation für deren Optimierung untersucht. Dabei wurden domänenspezifische Sprachen, maschinell erlernte Modelle und allgemeine Programmiersprachen betrachtet. Insbesondere die Anwendung auf Random Forests, eine Methode des klassischen maschinellen Lernens, war besonders erfolgreich. Hier konnten nicht nur Beschleunigungen von mehreren Größenordnungen erreicht werden, sondern auch gleich drei wichtige Erklärbarkeitsprobleme gelöst werden. Random Forests, die als nicht interpretierbare Black-Box-Modelle angesehen werden, können so semantisch aggregiert und verständlicher dargestellt werden. Das Resultat der Aggregation kann als semantisch äquivalentes White-Box-Modell angesehen werden. Die Lösung der Erklärbarkeitsprobleme ist beispielsweise in der Medizin oder im Bankensektor von enormer Bedeutung. Hier müssen automatisierte Entscheidungen immer erklärbar sein.

1 Einführung

Random Forests sind eine der beliebtesten Klassifikationsmethoden im klassischen maschinellen Lernen [Br01]. Sie finden Anwendung in den verschiedensten Bereichen, sei es im Bankensektor, im Gesundheitswesen [Ch21] oder im Web [Fa17], und sie helfen wichtige Entscheidungen zu automatisieren. Anstatt Kriterien für Entscheidungen manuell zu formalisieren, lernen Random Forests Muster automatisch und anhand von Beispieldaten. Auf diese Weise kann beispielsweise die Reaktion von Patienten auf eine Krebsbehandlung anhand ihrer DNA vorhergesagt werden [Ch21]. Obwohl Zusammenhänge wie dieser im Allgemeinen kaum oder gar nicht verstanden sind, können so gute Entscheidungen automatisch getroffen werden. Gleichzeitig birgt die Anwendung in diesen Bereichen ein Risiko, nämlich genau dann, wenn diese maschinell getroffenen Entscheidungen nicht mehr nachvollziehbar sind und man potentielle Fehler nicht erkennen kann. Je nach Anwendungsfall, zum Beispiel in der Medizin, kann dies aber enorm wichtig sein. Leider sind viele der erfolgreichsten Methoden im maschinellen Lernen in genau diesem Sinne nicht interpretierbar. Man nennt diese auch Black-Box-Modelle. Zu ihnen gehören auch die hier betrachteten Random Forests.

Durch holistische Aggregation von Random Forests ist es im Rahmen dieser Dissertation [Go21] gelungen deren Semantik zu aggregieren und in ein sehr viel verständlicheres und kompaktes Modell zu überführen. Dieses Resultat kann als Lösung des allgemeinen

[1] Englischer Titel der Dissertation: „Aggressive Aggregation - Domain-specific program optimization with Algebraic Decision Diagrams“

[2] Fakultät für Informatik, TU Dortmund, frederik.gossen@tu-dortmund.de

Erklärungsproblems für Random Forests gesehen werden. Die Aggregation erlaubt es sogar noch einen Schritt weiterzugehen und löst insgesamt gleich drei verschiedene Entscheidungsprobleme:

- Das allgemeine *Modell-Erklärungsproblem* [Gu19] fordert eine Erklärung des Klassifikationsmodells als Ganzes, hier des Random Forests. Das gelernte Modell soll so dargestellt werden, dass es für Experten verständlich ist und sie aus dem maschinell Gelernten Erkenntnisse ziehen können.
- Das *Klassen-Erklärungsproblem* [GS21] ist eine Reduktion des *Erklärungsproblems*. Es fordert eine Erklärung des Klassifikationsmodells bzgl. einer einzelnen Klasse, also zum Beispiel, ob ein Patient auf eine Krebsbehandlung besonders gut reagiert oder nicht.
- Das *Ergebnis-Erklärungsproblem* [Gu19] betrachtet eine konkrete Anwendung des gelernten Modells und fordert nur eine Erklärung für eine in einem bestimmten Fall getroffene Entscheidung.

Die ursprüngliche Zielsetzung dieser Arbeit war es, das Potential von algebraischen Entscheidungsdiagrammen (ADDs) [Ba97, Br86, Ak78] als Programmrepräsentation in Compilern zu untersuchen. Die Datenstruktur ist allgemein sehr gut verstanden und bekannt für ihre Optimalität in Größe und Tiefe [Ba97]. Im Kontext von Compilern entsprechen diese der Größe und der Laufzeit von Programmen, zwei klassische Ziele in der Programmoptimierung. Leider ist es nicht möglich allgemeine Programmiersprachen in diese Datenstruktur zu übersetzen. Beschränkt man allerdings die Domäne der betrachteten Programme auf solche, für die man Transformationen finden kann, so können die bekannten Algorithmen und Eigenschaften von ADDs ausgenutzt werden, um beeindruckende Laufzeitoptimierungen zu realisieren. Diese domänenspezifischen Programme können so teilweise um mehrere Größenordnungen beschleunigt werden. In Anwendungen mit enorm hohen Bandbreiten, kann dieser Effizienzgewinn enorm wertvoll sein [Fa17].

Im Rahmen dieser Dissertation wurde das Potential von ADDs als Programmrepräsentation in drei Domänen untersucht:

- Grafische und textuelle domänenspezifische Sprachen, die speziell für ADDs entworfen sind [St19, Go18] (siehe Abb. 1),
- Maschinell gelernte Modelle bzw. Programme am Beispiel von Random Forests [GS21, GMS20, GMS21] und
- Allgemeine Programmiersprachen am Beispiel der *while*-Sprache [Go19].

Im Folgenden werden wir genauer auf die Aggregation von Random Forests und die daraus resultierenden Vorteile eingehen. Wir werden sehen, wie selbst große Random Forests in ein einzelnes algebraisches Entscheidungsdiagramm überführt werden können und wie sukzessive Abstraktion den Random Forest gleichzeitig erklären und seine Auswertung enorm beschleunigen kann.

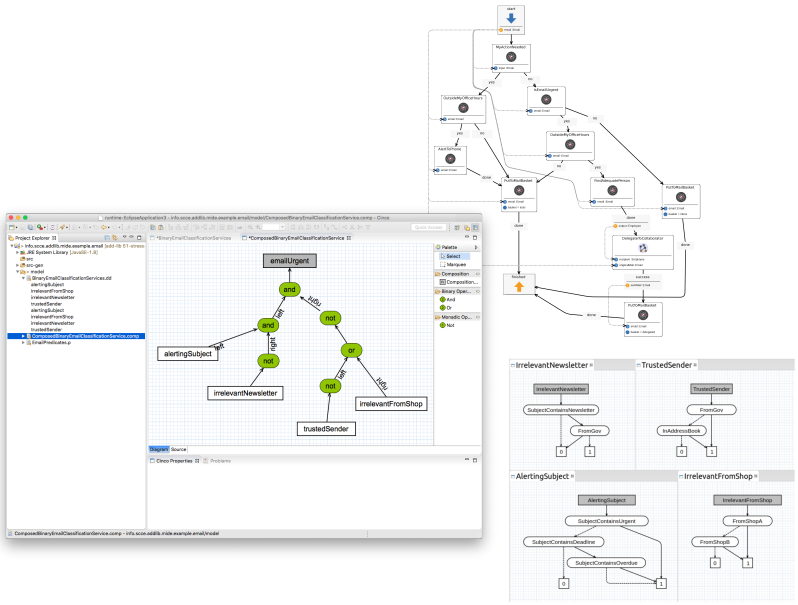


Abb. 1: Grafische ADD-basierte domänenspezifische Programmiersprachen und die darauf zugeschnittenen Entwicklungsumgebungen [St19].

2 Aggregation eines Random Forests

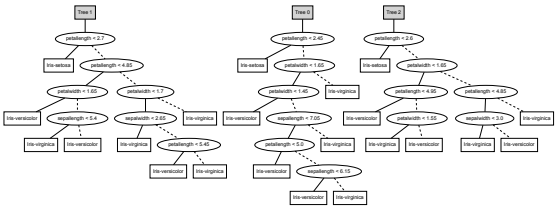


Abb. 2: Random Forest mit drei Bäumen, der auf dem Iris-Datensatz [Fi36] trainiert wurde. (41 Knoten)

Die erste Repräsentation ist der originale Random Forest wie er gelernt wurde [Br01]. Abbildung 2 zeigt einen solchen, der mit einer Standardmethode auf dem Iris-Datensatz [Fi36] trainiert wurde und im Folgenden als Beispiel dienen soll. Das Klassifikationsproblem des Datensatzes ist ein beliebtes Beispiel, um Methoden im maschinellen Lernen anschaulich darzustellen. Die Aufgabe ist es, anhand verschiedener Maße einer Irisblüte, die Korrekte von drei Spezies zu bestimmen. Für ein neues noch nicht klassifiziertes Beispiel sind also diese Maße gegeben und wir können jeden der drei Entscheidungsbäume von der Wurzel an auswerten, pro innerem Knoten den entsprechenden Nachfolger wählen und in den Blättern eine Vorhersage finden. Die Klassifikation des gesamten Random Forests

ist dann die am häufigsten gewählte Spezies, das Ergebnis des so genannten „majority votes“ [Br01]. [3]

Die Aggregation eines beliebig großen Random Forests zu einem einzelnen algebraischen Entscheidungsdiagramm (ADD) [Ba97] erfolgt in drei Schritten. Zunächst werden die Entscheidungsbäume des Random Forests elementweise in ADDs überführt. Von hier an kann die algebraische Natur und die Kompositionalität der ADD-basierten Repräsentation voll ausgenutzt werden um die vielen ADDs in einen einzelnen zu kondensieren. Der so resultierende ADD kann dann im dritten Schritt durch Abstraktion weiter optimiert werden, indem er zur Compile-Zeit maximal ausgerechnet wird und unerfüllbare Pfade eliminiert werden.

Es gibt zwei wesentliche Unterschiede zwischen den Entscheidungsbäumen des Random Forests (Abb. 2) und äquivalenten ADDs, die wir im ersten Schritt ableiten wollen. Während Prädikate zunächst in beliebiger und sogar verschiedener Reihenfolge in den unterschiedlichen Bäumen vorkommen können, fordern ADDs eine feste und einheitliche Reihenfolge ein. Darüber hinaus werden isomorphe Knoten in ADDs zu einem verschmolzen; jeder Knoten ist also eindeutig. Beide Invarianten lassen sich effizient umsetzen, sodass aus dem ursprünglichen Random Forest ein semantisch äquivalenter Forest aus ADDs wird.

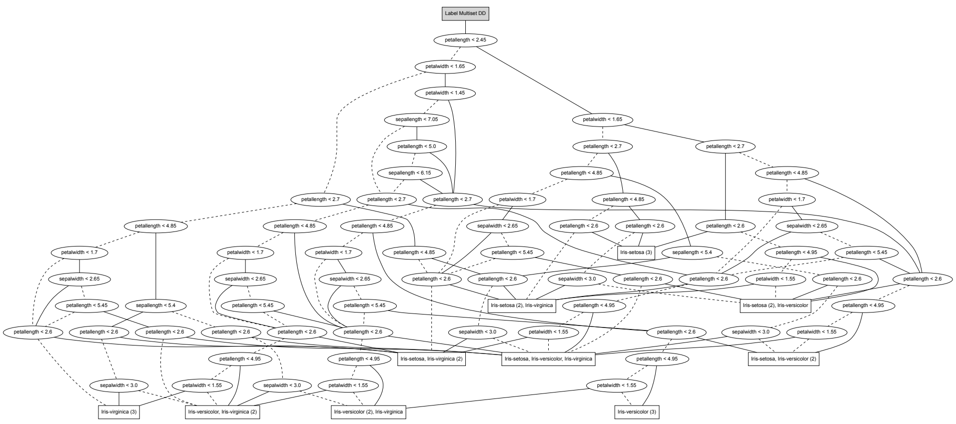


Abb. 3: Als Vektor-ADD aggregierter Random Forest. Dieser ADD ist semantisch äquivalent zu dem Random Forest in Abb. 2. (79 Knoten)

Um diese Menge von ADDs zu einem einzelnen zu aggregieren ist eine kompositionelle Darstellung notwendig. Die abstrakteste, noch kompositionale Darstellung, sind in diesem Fall Klassen-Vektoren, die eine ganzzahlige Dimension pro Klasse enthalten. Auf diese Weise lässt sich zählen, wie oft jede Klasse von einem der ursprünglichen Bäume gewählt wurde. Die Menge von ADDs lässt sich so einfach elementweise übersetzen, nämlich zu solchen ADDs, die anstatt einer einzelnen Klasse entsprechende Einheitsvektoren in ihren Blättern enthalten. Durch Summation dieser Klassen-Vektoren, lassen sich Entscheidungen der einzelnen ADDs auf natürliche Weise aggregieren. Die Datenstruktur erlaubt es

[3] Uneindeutigkeiten werden durch beliebige Priorität der Spezien gelöst.

aber auch, diese Aggregation auf der Metaebene der ADDs selbst durchzuführen. So werden die ADDs effektiv aufsummiert und der gesamte Random Forest wird in einem einzelnen, semantisch äquivalenten ADD dargestellt. Anstatt sämtliche Bäume des Random Forests auswerten zu müssen, reicht es nun, einen einzelnen ADD von seiner Wurzel an zu traversieren. Das Ergebnis ist ein Vektor, der die Wahlhäufigkeiten pro Klasse widerspiegelt. Die am häufigsten gewählte Klasse lässt sich daraus direkt ableiten. Abbildung 3 zeigt den aggregierten Vektor-ADD für den vorangegangenen Random Forest (Abb. 2).

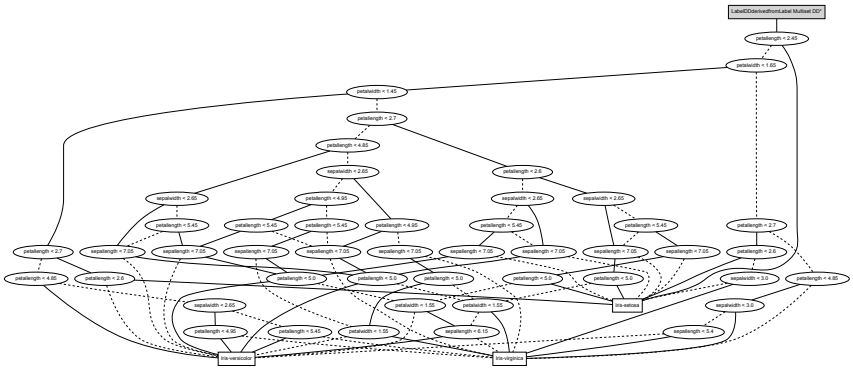


Abb. 4: Als Klassen-ADD aggregierter Random Forest. Dieser ADD wurde auf die Erfüllbarkeit seiner Pfade gefiltert und ist semantisch äquivalent zu dem Random Forest in Abb. 2 und dem ADD in Abb. 3. (50 Knoten)

Auch wenn die Klassen-Vektoren für die Aggregation unverzichtbar sind, weil sie die Kompositionalität der ADDs garantieren, so enthalten sie für die Auswertung des Modells unnötig viel Information. Tatsächlich sind wir nur an der am häufigsten gewählten Klasse interessiert, einer einfachen Abstraktion der Klassen-Vektoren. Wie zuvor, lässt sich auch diese Operation auf die Metaebene der ADDs heben und wir können den gesamten ADD so transformieren, dass seine Blätter direkt das gewünschte Ergebnis enthalten. Auf diese Weise ist es möglich den „majority vote“ schon zur Compile-Zeit auszurechnen. Der Effekt ist größer als er auf den ersten Blick scheint, denn die Abstraktion führt dazu, dass vorher verschiedene Blätter verschmolzen werden. Der ADD kollabiert so von seinen Blättern her und kann deutlich kleiner sein als zuvor.

Ein weiterer Aspekt für die Optimierung sind unerfüllbare Pfade. Weil ADDs symbolischer Natur sind und ihre Optimalitätsgarantien auf der Unabhängigkeit von Prädikaten beruhen, finden sich in dem aggregierten ADD unerfüllbare Pfade. So kann es passieren, dass sich auf einem Pfad widersprüchliche Bedingungen finden, wie zum Beispiel $petalwidth < 3$ und $\neg(petalwidth < 5)$. Diese Unerfüllbarkeiten können durch Intervallpropagation aus dem ADD eliminiert werden. Es ist so möglich den ADD weiter zu vereinfachen und seine Größe und Tiefe weiter zu reduzieren.

Abbildung 4 zeigt den aggregierten und semantisch äquivalenten ADD für den ursprünglichen Random Forest (Abb. 2). Diese Repräsentation aggregiert die Semantik auf eine Weise, die es erlaubt, den Random Forest (i) extrem schnell auszuwerten und (ii) seine Semantik besser zu verstehen.

3 Auswirkungen auf die Laufzeit

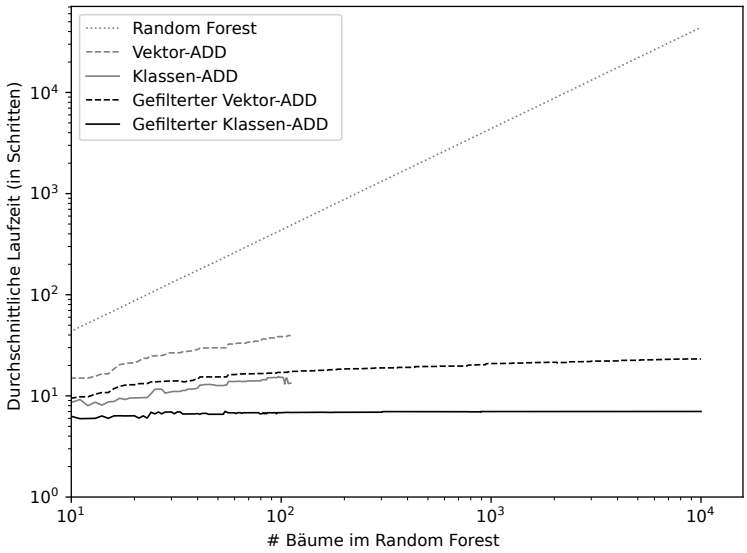


Abb. 5: Laufzeiten der verschiedenen Repräsentationen eines Random Forests mit bis zu 10.000 Bäumen (gemessen in Schritten).

Das ursprüngliche Ziel dieser Dissertation war es die Laufzeit von Random Forests zu reduzieren. Der aggregierte ADD erreicht dieses Ziel und kann die Laufzeit in einigen Fällen um mehrere Größenordnungen reduzieren.

Abbildung 5 zeigt die Auswirkungen der Repräsentation auf die Laufzeit für das vorangegangene Beispiel des Iris-Datensatzes. Anstatt eines Random Forests mit nur 3 Bäumen, betrachten wir hier Forests mit 1 bis 10.000 Bäumen. Die Laufzeit wird in Schritten durch die Datenstruktur gemessen, die für die Beispiele des Datensatzes durchschnittlich erforderlich sind.

Die durchschnittliche Laufzeit des originalen Random Forests wächst linear mit der Anzahl an Bäumen, weil jeder separat ausgewertet werden muss. Im Gegensatz dazu ist die Auswertung der aggregierten ADDs um bis zu drei Größenordnungen schneller. Für, auf unerfüllbare Pfade gefilterte, Klassen-ADDs scheint die Laufzeit sogar zu konvergieren.

Das Filtern auf Unerfüllbarkeit hat hier zweierlei Auswirkung: es hält die Datenstruktur klein und reduziert gleichzeitig deren Tiefe. Das führt einerseits zu schnelleren Laufzeiten, ermöglicht es andererseits aber überhaupt erst die Aggregation für größere Random Forests zu berechnen. Das ist auch der Grund dafür, dass die Laufzeit der ungefilterten ADDs nur bis zu einer Größe von 100 Bäumen untersucht wurde.

Die Ergebnisse des Iris-Datensatzes lassen sich auch auf anderen Datensätzen des UCI Machine Learning Repository [DG17] reproduzieren. Tabelle 1 zeigt durchschnittliche Laufzeiten für Random Forests mit 1.000 Bäumen auf anderen Datensätzen.

Datensatz	Originaler Random Forest	Aggregierter ADD
Balance Scale	8.014,12	7,73 (-99,90%)
Breast Cancer	13.020,03	17,11 (-99,87%)
Lenses	4.431,42	3,67 (-99,92%)
Iris	4.395,77	6,97 (-99,84%)
Tic-Tac-Toe	10.733,68	14,22 (-99,87%)
Vote	6.921,56	8,33 (-99,88%)

Tab. 1: Durchschnittliche Laufzeit für die Klassifikation (gemessen in Schritten). Betrachtet wird hier ein Random Forest mit 1.000 Bäumen, der jeweils auf einem Datensatz aus dem UCI Machine Learning Repository [DG17] gelernt wurde.

4 Lösungen der Erklärungsprobleme

Die Aggregation mittels algebraischer Entscheidungsdiagramme ist semantik-erhaltend und löst neben Laufzeitverbesserungen ein Problem einer weiteren Disziplin: *Erklärbarkeit* [Gu19]. Während Random Forests als Black-Box-Modelle nicht interpretierbar sind, kondensiert die Aggregation mit ADDs deren Semantik und stellt diese kompakt dar. Ähnlich wie Entscheidungsbäume können ADDs als White-Box-Modelle gesehen werden; sie sind also interpretierbar. Wenn man die sukzessive Abstraktion der ADDs fortsetzt, kann man sogar noch bessere Erklärungen für selektierte Aspekte des Modells finden.

Mit der ADD-basierten Aggregation ist es möglich gleich drei *Erklärbarkeitsprobleme* für Random Forests zu lösen [Gu19]:

- das allgemeine *Modell-Erklärungsproblem*,
- das *Klassen-Erklärungsproblem* und
- das *Ergebnis-Erklärungsproblem*.

Die Lösung des allgemeinen *Modell-Erklärungsproblems* wurde bereits in Abschnitt 2 diskutiert. Die Aggregation des Random Forests in einem einzelnen, redundanzfreien ADD ist ebenso einfach zu verstehen wie ein einzelner Entscheidungsbaum. Aus diesem Grund kann der präziseste und kompakteste ADD als Lösung des allgemeinen *Modell-Erklärungsproblems* für Random Forests gesehen werden. Abbildung 4 zeigt diese Modell-Erklärung für das Beispiel des Iris-Datensatzes.

Wenn man die sukzessive Abstraktion der ADDs fortführt, kann man auf ähnliche Weise die Lösung für ein spezielleres Erklärungsproblem finden: das *Klassen-Erklärungsproblem*. Hier wird die Erklärung nur für eine der betrachteten Klassen gefordert, also zum Beispiel für die Klasse *Iris-setosa*. Anstatt in den Blättern des ADD alle Klassen aufzulisten, unterscheidet man hier nur zwischen der Klasse *Iris-setosa* und allen anderen. Das Ergebnis ist ein zweiblättriger ADD bzw. ein BDD. Abbildung 6 zeigt diesen am Beispiel des Iris-Datensatzes. Die Semantik des ursprünglichen Random Forests kann so noch präziser für eine gewählte Klasse beschrieben werden.

Abb. 6: Als Klassen-BDD aggregierter Random Forest für die Klasse *Iris-setosa*. Bzgl. dieser Klasse ist der BDD semantisch äquivalent zu dem Random Forest in Abb. 2. (15 Knoten)

Das *Klassen-Erklärungsproblem* ist auch deshalb interessant, weil es den Klassifikationsprozess umdreht. Anstatt nach der korrekten Klasse für eine bestimmte Eingabe zu suchen, sucht man hier nach allen möglichen Eingaben für eine bestimmte Klasse. Diese Perspektivänderung kann zum Beispiel im Marketing hilfreich sein, weil man zwischen Kunden- und Produkt-Perspektiven wechseln kann [GMS20].

Das dritte Problem ist das *Ergebnis-Erklärungsproblem*. Es ist vor allem für die verantwortliche Nutzung von maschinellem Lernen wichtig und fordert die Erklärung in einem ganz konkreten Anwendungsfall [Gu19]. So können konkrete, automatisch getroffene Entscheidungen begründet werden, was insbesondere in der Medizin von enormer Bedeutung ist. Im Fall des Iris-Beispiels heißt das, dass wir eine Erklärung für die Klassifikation mit ganz konkreten Maßen fordern.

Die Lösung des vorangegangenen *Klassen-Erklärungsproblems* (Abb. 6) erlaubt es uns, auch das *Ergebnis-Erklärungsproblem* in zwei einfachen Schritten zu lösen:

Pfad-basierte Erklärung. Für die konkreten Maße einer Eingabe können wir den entsprechenden Klassen-BDD von seiner Wurzel an traversieren. Wenn wir die (negierten) Prädikate entlang dieses Pfades aufsammeln, erhalten wir eine präzise Erklärung. Deren Konjunktion ist eine hinreichende Bedingung für die getroffene Entscheidung. Beispielsweise erhalten wir für die Werte

$$petallength = 2.6,\ petalwidth = 1.5,\ petallength = 2.65,\ sepallength = 6.9$$

folgende hinreichende Bedingung:

$$petallength \geq 2.45 \wedge\ petalwidth \geq 1.45\ \wedge\ petalwidth < 1.65$$
$$\wedge\ petallength \geq 2.6\ \wedge\ petallength < 2.7 \wedge\ sepallength < 7.05.$$

Konjunktionen vereinfachen. Das Aufsammeln der Prädikate entlang eines Pfades kann erneut Redundanzen für das konkrete Beispiel zum Vorschein bringen. Das ist selbst dann der Fall, wenn der ADD bereits auf die Erfüllbarkeit seiner Pfade gefiltert wurde. Diese können nun genutzt werden, um die Konjunktion weiter zu vereinfachen, indem man nur die stärksten Prädikate aufsammelt. In dem vorangegangenen Beispiel ist $petallength \geq 2.45$ redundant, weil es von dem stärkeren Prädikat $petallength \geq 2.6$ impliziert wird. Auf diese Weise erhält man eine minimale Lösung für das *Ergebnis-Erklärungsproblem*.

5 Schlussfolgerungen

Durch die semantik-erhaltende Aggregation von Random Forest zu einem einzelnen algebraischen Entscheidungsdiagramm, ist es möglich deren Semantik zu kondensieren. Die semantik-erhaltende Transformation lässt sich algebraisch elegant definieren [GS21] und erhält die Kompositionalität zunächst vollständig. Durch eine Reihe nicht-kompositionaler Abstraktionen ist es dann möglich bestimmte Aspekte des Modells hervorzuheben oder seine Größe und Tiefe (bzw. Laufzeit) zu optimieren.

Auf diese Weise werden gleich drei Erklärungsprobleme für diese Methode des maschinellen Lernens gelöst: das *Modell-Erklärungsproblem*, das *Klassen-Erklärungsproblem* und das *Ergebnis-Erklärungsproblem*. Alle drei Lösungen sind von enormer Bedeutung in den verschiedensten Anwendungsgebieten im maschinellen Lernen, beispielsweise in der Medizin [Ch21] oder im Bankensektor.

Gleichzeitig erlaubt die radikale Aggregation es, den Random Forest deutlich schneller auswerten zu können, was in Anwendungen mit hohen Bandbreiten von enormer Bedeutung ist [Fa17]. Anstatt jeden Entscheidungsbaum des Random Forests einzeln auswerten zu müssen, reicht es, einen einzelnen aggregierten ADD zu traversieren. So ist es in mehreren Anwendungsfällen möglich, die gleiche Semantik um mehrere Größenordnungen schneller auswerten zu können.

Literaturverzeichnis

[Ak78] Akers, S. B.: Binary Decision Diagrams. IEEE Trans. Comput., 27(6):509–516, 1978.

[Ba97] Bahar, R.I.; Frohm, E.A.; Gaona, C.M.; Hachtel, G.D.; Macii, E.; Pardo, A.; Somenzi, F.: Algebric Decision Diagrams and Their Applications. Formal Methods in System Design, 10, 1997.

[Br86] Bryant, Randal E.: Graph-Based Algorithms for Boolean Function Manipulation. IEEE Trans. Comput., 35(8):677–691, 1986.

[Br01] Breiman, Leo: Random Forests. Machine Learning, 45(1), 2001.

[Ch21] Chowell, D., Yoo SK. Valero C. et al.: Improved prediction of immune checkpoint blockade efficacy across multiple cancer types. Nature Biotechnology, 2021.

[DG17] Dua, Dheeru; Graff, Casey: , UCI Machine Learning Repository. `http://archive.ics.uci.edu/ml`, 2017. Accessed: 2020-02-15.

[Fa17] Facebook: , Evaluating boosted decision trees for billions of users. `https://code.fb.com/ml-applications/evaluating-boosted-decision-trees-for-billions-of-users`, 2017. Accessed: 2019-06-11.

[Fi36] Fisher, R. A.: The Use of Multiple Measurements in Taxonomic Problems. Annals of Eugenics, 7(7):179–188, 1936.

[GMS20] Gossen, F.; Margaria, T.; Steffen, B.: Towards Explainability in Machine Learning: The Formal Methods Way. IT Professional, 22(04):8–12, 2020.

[GMS21] Gossen, Frederik; Margaria, Tiziana; Steffen, Bernhard: Formal Methods Boost Experimental Performance for Explainable AI. IT Professional, 23(6):8–12, 2021.

[Go18] Gossen, Frederik; Margaria, Tiziana; Murtovi, Alnis; Naujokat, Stefan; Steffen, Bernhard: DSLs for Decision Services: A Tutorial Introduction to Language-Driven Engineering. In (Margaria, Tiziana; Steffen, Bernhard, Hrsg.): Leveraging Applications of Formal Methods, Verification and Validation. Modeling. Springer International Publishing, S. 546–564, 2018.

[Go19] Gossen, Frederik; Jasper, Marc; Murtovi, Alnis; Steffen, Bernhard: Aggressive Aggregation: a New Paradigm for Program Optimization. CoRR, abs/1912.11281, 2019.

[Go21] Gossen, Frederik: Aggressive Aggregation - (Domain-specific) Program Optimisation with Algebraic Decision Diagrams. LS 05 Programmiersysteme, 2021.

[GS21] Gossen, F.; Steffen, B.: Algebraic Aggregation of Random Forests: Towards Explainability and Rapid Evaluation. International Journal on Software Tools for Technology Transfer, 2021.

[Gu19] Guidotti, Riccardo; Monreale, Anna; Ruggieri, Salvatore; Turini, Franco; Giannotti, Fosca; Pedreschi, Dino: A Survey of Methods for Explaining Black Box Models. ACM Comput. Surv., 51(5):93:1–93:42, 2019.

[St19] Steffen, Bernhard; Gossen, Frederik; Naujokat, Stefan; Margaria, Tiziana: Language-Driven Engineering: From General-Purpose to Purpose-Specific Languages. In (Steffen, Bernhard; Woeginger, Gerhard, Hrsg.): Computing and Software Science: State of the Art and Perspectives. Springer International Publishing, S. 311–344, 2019.

Frederik Gossen wurde am 5. Februar 1992 in Herdecke, Deutschland, geboren. Seit September 2015 promovierte er in der Informatik in Kooperation mit Lero, dem irischen Zentrum für Software Engineering Research, an den Universitäten *University of Limerick*, Irland, und *Technische Universität Dortmund*, Deutschland. In seiner Forschung untersuchte er das Potential von algebraischen Entscheidungsdiagrammen für die Programmoptimierung in verschiedenen Domänen, insbesondere im maschinellen Lernen. Seit April 2020 arbeitet er bei Google an Compilern und Programmoptimierung für maschinelles Lernen, zunächst in München, Deutschland, dann in New York, USA.

Hochqualitativ Verifikationsablauf für Heterogene Systeme auf Basis Virtueller Prototypen[1]

Muhammad Hassan[2]

Abstract: In dieser Dissertation werden mehrere neuartige Ansätze entwickelt, die verschiedene Verifikationsaspekte abdecken, um den modernen, auf *Virtuellen Prototypen* (VP)-basierten, Verifikationsablauf stark zu verbessern. Die Beiträge sind im Wesentlichen in vier Bereiche unterteilt: Der erste Beitrag führt eine neue Verifikationsperspektive für VPs ein, indem er *Metamorphic Testing* (MT) verwendet, da im Gegensatz zu modernen VP-basierten Verifikationsabläufen keine Referenzmodelle/-werte für die Verifikation benötigt werden. Der zweite Beitrag schlägt hochqualitative Methoden zum Schließen der Code-Abdeckung in modernen VP-basierten Verifikationsabläufen vor, indem er Mutationsanalyse und stärkere Abdeckungsmetriken wie Datenfluss-Abdeckung berücksichtigt. Der dritte Beitrag besteht aus einer Reihe hochqualitativ, neuartiger, systematischer und leichtgewichtigen funktionalen Methoden zur Verbesserung der relevanten Abdeckungsmetriken. Der vierte und letzte Beitrag dieser Arbeit sind neuartige Ansätze, die eine frühzeitige Sicherheitsvalidierung von VPs ermöglichen. Alle Ansätze werden im Detail vorgestellt und ausführlich mit mehreren Experimenten evaluiert, die ihre Effektivität durch einen hochqualitativ VP-basierten Verifikationsfluss für heterogene Systeme deutlich machen.

1 Einführung

Internet Der Dinge (engl. *Internet-of-Things*, IOT) und intelligente Geräte sind ein Hauptbeispiel für heterogene *System-On-Chips* (SOCs), die aus zwei Teilen bestehen: (1) Mixed-Signal *Hardware* (HW), wo die analoge Welt auf die digitale Welt trifft, (2) und *Software* (SW), die unsichtbare Schicht, die uns mit der physischen Realität verbindet. Heterogene SOCs gehören zu dem am schnellsten wachsenden Marktsegmenten in der Elektronik- und Halbleiterindustrie. Angetrieben von den Wachstumschancen in verschiedenen Anwendungsbereichen passen sich viele Halbleiterhersteller an und verlagern ihren Schwerpunkt von separaten *Integrierten Schaltungen* (engl. *Integrated Circuits*, ICs), die nur eine Funktion erfüllen, hin zu einer stärker integrierten Lösung für *Hochfrequenz* (engl. *Radio Frequency*, RF) und leistungsstarke *Analog/Mixed-Signal* (AMS) Designs. Diese Verlagerung hat zwar zu einer hohen Leistungsfähigeit und sehr effizienten Geräte mit geringem Platzbedarf geführt, z.B. der Apple M1 SOC, aber es wurde dadurch den Aufwand für die Entwicklung und Verifikation dieser hochkomplexen Bauelemente erheblich gesteigert, um die notwendigen Anforderungen bezüglich *Time-To-Market* (TTM) zu erreichen.

Eine große Herausforderung in dieser Hinsicht ist die Abhängigkeit von HW und SW. Herkömmlicherweise wurden HW und SW isoliert entwickelt und trafen erst in den späten

[1] Englischer Titel der Dissertation: "Enhanced Modern Virtual Prototype based Verification Flow for Heterogeneous Systems"

[2] Deutsches Forschungszentrum für Künstliche Intelligenz (DFKI), muhammad.hassan@dfki.de

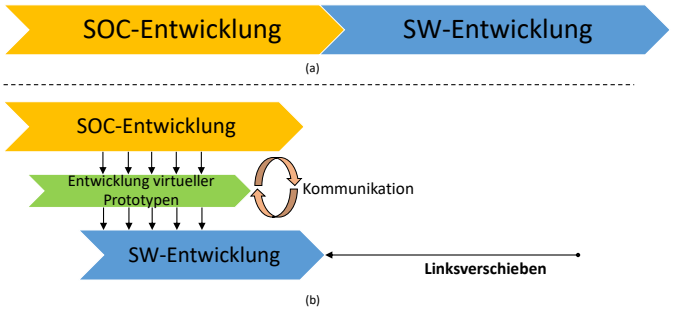


Abb. 1: Frühe SW-Entwicklung unter Nutzung des Linksverschiebungskonzepts

Integrations- und Testphasen aufeinander. Infolgedessen bestand immer eine sequentielle Abhängigkeit zwischen den HW- und SW-Entwicklungsphasen, wie in Abb. 1 (a) dargestellt ist. Daher konnte die SW erst dann richtig getestet werden, wenn die ersten Silizium-Prototypen des SOC verfügbar waren. Insbesondere HW-abhängige SW wie Gerätetreiber und Low-Level-Kernel-Code konnten erst geschrieben werden, nachdem das Siliziumdesign abgeschlossen war.

Eine weitere Herausforderung bei der Verifikation heterogener SOCs ist die langsame gemeinsame Simulationsgeschwindigkeit von *Register Transfer Level* (RTL) und SPICE-Modellen (Simulation Program with Integrated Circuit Emphasis) für den digitalen und den analogen/RF-Teil des SOCs [Ba10]. Traditionell war die Analog/RF-Verifikationsmethodik von Natur aus ad-hoc und diese IPs wurden immer von separaten Teams verifiziert. Sie wurde durch gezielte Tests über Sweeps, Ecken und Monte-Carlo-Analysen gesteuert. Diese vorverifizierten analogen, RF- und digitalen IPs wurden später in ein überwiegend digitales SOC-Design integriert und SW wurde darauf ausgeführt, um zu testen, ob alles wie erwartet funktioniert. Die gemeinsame Simulation ist zwar langsam, wird aber wegen ihrer hohen Genauigkeit immer noch als goldener Standard angesehen und kann nicht ignoriert werden. Für Simulationen auf Chipebene ist sie jedoch zu langsam, es sei denn, sie wird extrem selektiv eingesetzt.

Zusammenfassend lässt sich sagen, dass der erfolgreiche Co-Entwurf von sicheren multidisziplinären heterogenen SOCs, die enge Wechselwirkungen zwischen HW/SW-Systemen und ihrer analogen physikalischen Umgebung aufweisen, eine große Herausforderung darstellt. Je kürzer TTM wird, desto wichtiger wird die Fähigkeit, komplexe heterogene SOCs zu modellieren und zu simulieren, bei denen digitale HW/SW funktional mit AMS-IPs, d.h. HF-Schnittstellen, Leistungselektronik, Sensoren und Aktoren, verflochten sind. Wenn solche Modelle auf Gesamtsystem- und Architekturebene so früh wie möglich im Entwurfszyklus zur Verfügung stehen, werden die Probleme bei der Architekturexploration und beim Entwurf sowie den aufdeckung von Sicherheitslücken drastisch reduziert.

2 Entwurf und Verifizierung auf der Elektronischen Systemebene

In diesem Zusammenhang hat das Aufkommen *Virtueller Prototypen* (engl. *Virtual Prototypes*, VPs) auf der Abstraktionsebene der *Elektronischen Systemebene* (engl. *Electronic*

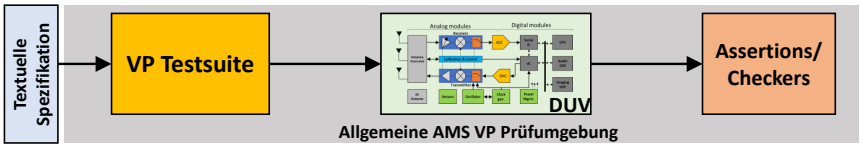


Abb. 2: Allgemeine AMS VP Prüfumgebung

System Level, ESL) den Entwurf und die Verifikation heterogener SOCs in vielerlei Hinsicht modernisiert [GD10, HGD20, Ha18, Ha17]. Der Grundgedanke besteht darin, ein abstraktes Referenzmodell des SOC auf Basis der schriftlichen Spezifikationen zu erstellen. Als Ergebnis wird eine ausführbare Beschreibung zur Verfügung gestellt, die als goldene Referenz für die HW- und SW-Entwicklung verwendet wird. Das *virtuelle Prototyping* bietet SW-Entwicklern und Systemarchitekten eine Umgebung für die SW-Entwicklung, die Betrachtung von Entwurfsalternativen bezüglich der Architektur oder das HW/SW-Co-Design. In einem vollständigen SOC-Entwurfsablauf liegt das virtuelle Prototyping zwischen der Funktions- und der Implementierungsebene. Auf dieser Abstraktionsebene nutzt das virtuelle Prototyping das *Linksverschiebungskonzepts* aufgrund seiner frühen Verfügbarkeit Abb. 1(b), d.h. der HW-Architekturentwurf und der SW-Entwicklungsfluss werden parallel und verschachtelt durchgeführt. Für den ESL-Entwurf ist die C++-basierte Systemmodellierungssprache SystemC zusammen mit *Transaction Level Modeling* (TLM)-Techniken (IEEE-Standard 1666) [In06] und deren Mixed-Signal-Erweiterung SystemC AMS [Ba13] mit überwiegend *Timed Data Flow* (TDF)-Modellentwicklung der Stand der Technik. Zu den Hauptvorteilen von VPs gehören die viel frühere Verfügbarkeit sowie die deutlich schnellere Simulationsgeschwindigkeit im Vergleich zu RTL-Modellen (für digitale) und SPICE-Level-Modellen (für AMS). Als Referenz für die (frühe) Entwicklung von eingebetteter SW und die HW-Verifikation ist die funktionale Korrektheit und Sicherheitsvalidierung von VPs sehr wichtig. Daher werden sowohl der gesamte VP als auch seine einzelnen Komponenten, d.h. Hochgeschwindigkeits-RF, AMS und digitale IPs, einer strengen Funktionsverifizierung unterzogen.

Trotz der jüngsten Fortschritte bei der formalen Verifikation von System-C/AMS-Modellen (siehe z.B. [GD10, Le16, He16]), ist die simulationsbasierte Verifikation dank ihrer Skalierbarkeit und einfachen Handhabung immer noch die Methode der Wahl für SystemC/AMS-basierte VP. Eine allgemeine simulationsbasierte AMS VP Verifikationsumgebung ist in Abb. 2 dargestellt. Sie folgt den Prinzipien des *gerichteten Testens* (engl. *Directed Testing*). Grundsätzlich werden die *textuellen Spezifikationen* verwendet, um manuell eine Reihe von Stimuli (*VP Testsuite*) zu erstellen, die auf das AMS DUV (das entweder ein ganzer VP, eine Reihe von Komponenten oder eine einzelne Komponente sein kann) angewendet werden, um bestimmte Szenarien zu testen. Für jeden Stimulus wird das tatsächliche Verhalten im Vergleich zum erwarteten Verhalten geprüft (z.B. spezifiziert durch Referenzausgaben in Form von *Assertions/Prüfern* oder zeitlichen Eigenschaften). Wenn die Assertions/Prüfer fehlschlagen, wird das DUV für fehlerhaft erklärt.

Dieser allgemeine simulationsbasierte Verifikationsablauf (gerichtetes Testen) ist zwar für die anfängliche Verifikation eines einfacheren DUV wichtig, aber für komplexe Entwürfe und eine gründliche Verifikation ist er nicht effektiv. Daher wird heutzutage ein moderner VP-Verifikationsablauf mit verschiedenen Methoden zur Ergänzung des allgemeinen

Verifikationsablaufs verwendet. Dies ist in Abb. 3 dargestellt. Die x-Achse zeigt vier farbige Balken, die die Methoden des modernen Verifikationsflusses vom Start bis zur Abnahme darstellen, und die y-Achse stellt die entsprechende Verifikationsqualität dar, die durch jeden Satz von Methoden erreicht wird. Der Grundgedanke ist, mit dem gerichteten Testen zu beginnen, wie zuvor beschrieben. Dies wird als grauer Farbbalken dargestellt. Die Verifikationsqualität ist in dieser Phase sehr schlecht, da die Teststimuli nur bestimmte Szenarien verifizieren. Anschließend wird die anfängliche Menge an Stimuli aus dem gerichteten Testen zusätzlich zu den Techniken des eingeschränkten Zufalls für Regressionstests verwendet. Regressionstests erfassen effektiv (1) Fehler, die während der DUV-Entwicklung eingeführt wurden, (2) und die Code-Abdeckung (ausgeübte Codezeilen) als Grundlinien und Trends. Dies wird durch einen orangefarbenen Balken dargestellt. Die durch Regressionstests erzielte überprüfungsqualität ist besser als bei gezielten Tests, aber immer noch schlecht, da der Schwerpunkt auf der Entwicklung von Stimuli liegt. Anschließend wird der Verifikationsfluss unter Nutzung der Code-Abdeckungs-Basislinien so umgestellt, dass eine geschlossene Code-Abdeckung erreicht wird und somit die Verifikationsqualität steigt. . Dies wird durch den blauen Farbbalken dargestellt. Am Ende wird die Abdeckungsmetrik auf funktionale Abdeckung umgestellt (ausgeübte Funktionen einer DUV) und die Stimuli werden verbessert, um einen Abschluss zu erreichen (gelber Farbbalken). An diesem Punkt wird die Qualität der Verifizierung als ausreichend angesehen und die Verifizierung wird abgeschlossen. Der moderne VP-Verifikationsablauf weist jedoch noch Schwächen auf, die zu qualitativ schlechten Teststimuli und VP führen. Die Schwächen werden im Folgenden kurz erörtert:

1. Bei Regressionstests ist die Verfügbarkeit von Referenzmodellen für den Vergleich der Ergebnisse immer noch eine große Herausforderung. Bei komplexen DUVs ist ein erheblicher Aufwand erforderlich, um das Referenzverhalten in einer ausführbaren Weise zu spezifizieren. Insbesondere die Interaktion von analogen Designs und digitaler Logik hat in modernen AMS SOCs erheblich zugenommen. Die Formalisierung solcher Interaktionen ist nicht trivial und sehr zeitaufwändig [HGD21a, HGD21b].
2. Die bestehenden Methoden zur Schließung der Code-Abdeckung sind zwar gut, aber in zweierlei Hinsicht unzureichend. Erstens verwenden sie nur schwache Abdeckungsmetriken, z.B. Anweisungsabdeckung und Verzweigungsabdeckung usw. Die schwachen Abdeckungsmetriken sind unempfindlich gegenüber varia-

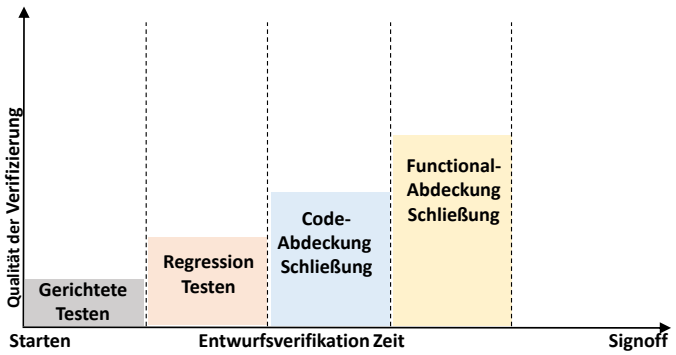


Abb. 3: Vier Stufen des modernen simulationsbasierten VP-Verifikationsablaufs

blen Interaktionen in dem VP, d.h. wie sich die in einem Teil des VP durchgeführten Berechnungen auf die anderen Teile auswirken. Zweitens berücksichtigen sie nicht die HW/SW-Interaktionen, die später zu einer großen Anzahl von IP-Integrationsproblemen führen.

3. Für eine gründliche Verifikation des DUV ist eine Verfolgung des Verifikationsfortschritts erforderlich. Bei der Verifikation digitaler Entwürfe ist insbesondere die funktionale Abdeckung die Metrik, die für diese Aufgabe verwendet wird, da sie es ermöglicht zu messen, ob alle Spezifikationen des Entwurfs verifiziert wurden. Während die funktionale Abdeckung in digitalen Design sehr gut verstanden wird, ist dies bei AMS nicht der Fall [Ha19], da kontinuierliche Signale und ihre Veränderung über die Zeit viel schwieriger zu erfassen sind.

4. Sicherheit ist heutzutage eines der wichtigsten Themen bei der Entwicklung eingebetteter Systeme. Die meisten Strategien zur Sicherung eingebetteter Systeme werden in SW implementiert. Eine potenzielle Hintertür in der HW, die nicht privilegierter SW den Zugriff auf vertrauliche Daten ermöglicht, macht jedoch selbst eine perfekt gesicherte SW unbrauchbar. Da der zugrundeliegende SOC nach der Implementierung nicht mehr gepatcht werden kann, ist es sehr wichtig, SOC-Hardware-Sicherheitsprobleme bereits in der Entwurfsphase zu erkennen und zu korrigieren.

Im nächsten Abschnitt werden die Beiträge dieser Arbeit erörtert, die die Qualität des modernen VP-Verifikationsablaufs stark verbessern. Diese Arbeit ist eine Zusammenfassung der Dissertation [Ha21].

3 Beitrag der Dissertation

In dieser Arbeit werden mehrere neue Ansätze und Methoden vorgeschlagen, um eine hochqualitativ VP Verifikation durchzuführen. Die Beiträge werden, wie in Abb. 4 gezeigt, nach den gezielten Tests in vier Hauptbereichen vorgeschlagen. Zunächst wird in dieser Arbeit eine neue Verifikationsperspektive für die VP-Verifikation vorgeschlagen: Metamorphes Testen zur effektiven Verifikation des VP ohne Referenzmodelle. Dies wird in Abb. 4 durch einen grünen Farbbalken dargestellt. Dann schlägt diese Arbeit Methoden, die eine starke Verbesserungen von bis zu 30% bei der Code-Abdeckung und Methoden zur Schließung der funktionalen Abdeckung vor (blaue bzw. gelbe Farbbalken). Schließlich wird eine Sicherheitsvalidierung des VPs eingeführt (pinkfarbener Balken), um Sicherheitsprobleme bereits in der Entwurfsphase zu erkennen. In diesem Zusammenhang konzentrieren wir uns auf digitale Systeme, die in den letzten Jahren kompromittiert wurden, z.B. Sicherheitslücken in SOCs, die Intels Mikroprozessor-IPs (Meltdown- und Spectre Schwachstellen) und Actel ProASIC3 IPs (JTAG-Schwachstelle) verwenden. Daher kann die frühzeitige Erkennung solcher Sicherheitslücken für das SOC entscheidend sein. Ein detaillierterer überblick über die Beiträge der Dissertation ist auf der linken Seite von Abb. 5 zu sehen. Die vier Bereiche der Beiträge verwenden allgemeine VP-Verifikationsumgebungen als Basis und bauen darauf deutlich erweiterte Verifikationsumgebungen auf:

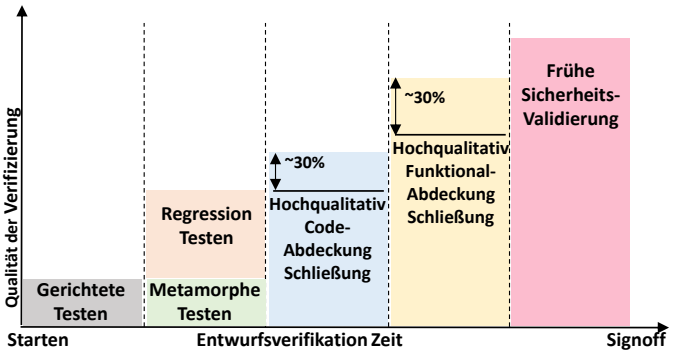


Abb. 4: Vorgeschlagener hochqualitativ VP-Verifikationsablauf

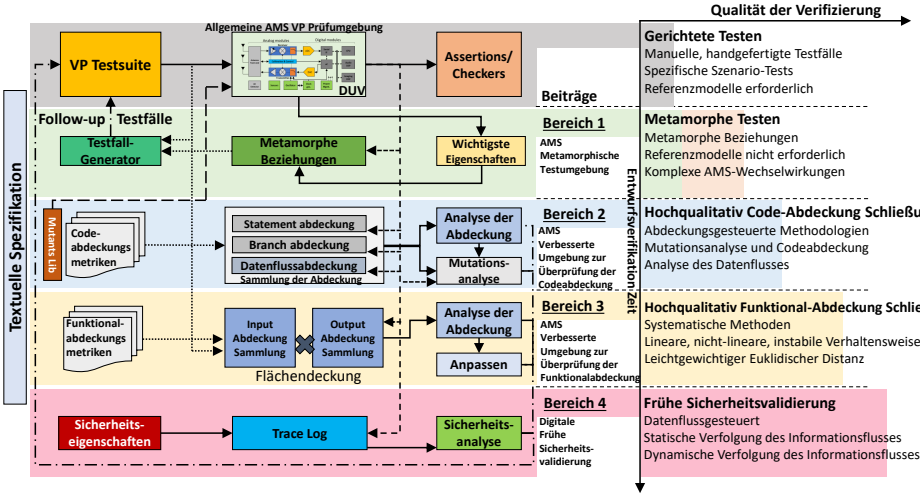


Abb. 5: Beiträge der Dissertation: Hochqualitativ VP-Verifikationsablauf

1. AMS metamorphische Testumgebung
2. AMS hochqualitativ Umgebung zur Überprüfung der Code-Abdeckung
3. AMS hochqualitativ Umgebung zur Überprüfung der Funktional-Abdeckung
4. Digitale frühe Sicherheitsvalidierung

Jeder Beitragsbereich erhöht die Verifikationsqualität des VP erheblich. Dies spiegelt sich auf der x-Achse in Abb. 5 (auf der rechten Seite) wider. Die y-Achse stellt die entsprechende Verifikationsumgebung und -methodik vom Start bis zur Abnahme im VP-Verifikationsablauf dar. Die Beiträge, die zu einer qualitativ hochwertigen Testbench und einem gründlich verifizierten VP führen, werden im Folgenden näher erläutert:

Beitragsbereich 1 - AMS Metamorphische Testumgebung Der erste Beitrag dieser Arbeit behebt die erste Schwäche - die Verfügbarkeit von Referenzmodellen. Der Beitrag führt eine neue Verifikationsperspektive für VPs ein, indem er *Metamorphisches Testen* (MT) verwendet. MT verifiziert ein DUV, indem es die bereits verfügbaren Testfälle

aus gerichteten oder Regressionstests (als Basisteststimuli bezeichnet) berücksichtigt, um neue Testfälle (als Folgeteststimuli bezeichnet) zu generieren, ohne dass ein Referenzmodell benötigt wird. MT tut dies, indem es einfach vorhandene Testfälle mit neu erstellten Testfällen in Beziehung setzt. Das zentrale Element von MT ist ein Satz von *Metamorphischen Relationen* (MR), die die Beziehung zwischen den Eingaben und Ausgaben aufeinanderfolgender DUV-Ausführungen anhand von Kerneigenschaften des DUV beschreiben. Da für MT keine Referenzmodelle erforderlich sind, ergänzt es wirksam die Regressionstests. Infolgedessen gewährleisten die identifizierten MRs die Korrektheit des DUV über mehrere DUV-Versionen hinweg während des Entwurfs und der Verifikation. Die vorgeschlagene AMS Metamorphische Testumgebung ist in Abb. 5 grün hinterlegt. Sie besteht aus (1) einem Testfallgenerator, (2) metamorphen Beziehungen und (3) Kerneigenschaften des DUV. Die Idee auf hoher Ebene ist die Erstellung von MRs unter Verwendung der Kerneigenschaften des DUV. Anschließend werden die MRs im Testfallgenerator Modul verwendet, um neue Testfälle zu erstellen, die Fehler aufdecken können.

Daher stellen wir in dieser Arbeit einen neuen MT-basierten Verifikationsansatz vor, um das lineare und nichtlineare Verhalten von HF-Verstärkern auf Systemebene zu verifizieren. Für die Klasse der HF-Verstärker identifizieren wir hochwertige MRs zur Verifizierung des linearen und nichtlinearen Verhaltens. Darüber hinaus gehen wir über reine Analog/RF-Systeme hinaus, d.h. wir erweitern MT, um AMS-Systeme zu verifizieren. Als anspruchsvolles AMS-System betrachten wir eine industrielle PLL und entwickeln eine Reihe von hochwertigen MRs. Diese MRs ermöglichen die Verifizierung des PLL Verhaltens auf Komponenten- und Systemebene zu überprüfen. Daher wird in dieser Arbeit ein MT-basierter Verifikationsansatz vorgeschlagen, der sowohl das Analog-Analog-, Analog-Digital-, Digital-Analog- als auch das Digital-Digital-Verhalten berücksichtigt.

Beitragsbereich 2 - AMS hochqualitativ Umgebung zur überprüfung der Code-Abdeckung Als zweiter Beitrag werden in dieser Arbeit Methoden zur Schließung der Code-Abdeckung mit hoher Qualität in modernen VP-basierten Verifikationsabläufen vorgeschlagen. Die Methoden erreichen eine signifikante Verbesserung der Verifikationsqualität um bis zu 30%. Die vorgeschlagene AMS hochqualitativ Umgebung zur Überprüfung der Code-Abdeckung ist in Abb. 5 in blauem Hintergrund dargestellt. Sie umfasst verschiedene Code-Abdeckungsmetriken und eine neuartige Abdeckungsanalyse. Darüber hinaus nutzt sie die Mutationsbibliothek und die Mutationsanalyse, um einen qualitativ hochwertigen VP zu erreichen. VP-basierte Designs ermöglichen das HW/SW-Co-Design und damit die softwaregetriebene Verifikation (engl: *Software Driven Verification*, SDV), die den Gesamtaufwand für die IP-Integration und -Verifikation erheblich zu reduzieren verspricht. Mit Hilfe von SystemC VPs können SW-Tests zur Verifikation der (neuen) integrierten IP-Blöcke und der HW/SW-Integration in einer frühen Entwurfsphase entwickelt und in den nachfolgenden Schritten wiederverwendet werden. Zu diesem Zweck schlagen wir in dieser Arbeit eine neuartige qualitätsgesteuerte Methodik vor, die auf einer Mutationsanalyse basiert. Durch die übertragung der wichtigsten Konzepte der mutationsbasierten Qualifizierung in den Kontext des SDV ist unsere Methodik in der Lage, ernsthafte Qualitätsprobleme in den SW-Tests zu erkennen. Das Herzstück ist eine neuartige Konsistenzanalyse, die die Abdeckung des IP in der HW/SW-Cosimulation auf leichtgewichtige Weise misst und diese Abdeckung mit den SW-Testergebnissen in Beziehung setzt, um ein

klares Feedback zu geben, wie die Qualität der Tests weiter verbessert werden kann. Obwohl dies ein notwendiger Schritt ist, haben Anweisungs- und Verzweigungsabdeckung im Zusammenhang mit SDV einige bekannte Einschränkungen hinsichtlich ihrer Fähigkeit, Fehler zu erkennen und die Gründlichkeit der Verifikation wiederzugeben. Sie sind unzureichend, wenn es darum geht, die Wechselwirkungen zwischen verschiedenen Elementen (Variablen) in einem VP zu berücksichtigen.

In dieser Hinsicht verbessert das Datenfluss-basierte Testen (engl: *Data Flow Testing*, DFT) die Qualität der Verifikation, indem es berücksichtigt, wie ein syntaktisches Element die Berechnung eines anderen beeinflussen kann. Daher schlagen wir in dieser Arbeit vor, DFT für SystemC/AMS VPs anzuwenden, da die modernen VPs nicht mehr nur digital sind, sondern multifunktionale AMS SOCs sind. Wir entwickeln zunächst eine Reihe von SystemC/AMS-spezifischen Abdeckungskriterien für DFT. Dies erfordert die Berücksichtigung (1) der SystemC-Semantik der Verwendung von nichtpräemptivem Thread-Scheduling mit Shared-Memory-Kommunikation und ereignisbasierter Synchronisation, (2) der SystemC-AMS-Semantik des Signalflusses im Allgemeinen und zeitgesteuerter Datenflussmodelle im Besonderen. Anschließend wird erläutert, wie die Datenfluss-Abdeckung für einen gegebenen VP mit einer Kombination aus statischen und dynamischen Analysetechniken automatisch berechnet werden kann. Die überdeckungsergebnisse liefern klare Vorschläge für den Verifikationsingenieur, neue Testfälle hinzuzufügen, um das Abdeckungsergebnis zu verbessern.

Beitragsbereich 3 - AMS hochqualitativ Umgebung zur Überprüfung der Funktional-Abdeckung Der dritte Beitrag dieser Arbeit ist eine Reihe von hochqualitativen Methoden zum Schließen der funktionalen Abdeckung in modernen VP-Verifikationsabläufen, die die Qualität der Verifikation um bis zu 30% erhöhen. Die AMS hochqualitativ Umgebung zur Überprüfung der Funktional-Abdeckung ist in Abb. 5 gelb unterlegt dargestellt. Sie umfasst Überdeckungssammelbehälter am Eingang und Ausgang des DUV sowie eine Überdeckungsanalyse. Darüber hinaus wird ein Anpassungsmodul eingeführt, um die Stimuliererzeugung automatisch zu steuern.

In dieser Arbeit wird ein Verifikationsansatz mit funktionaler Abdeckung als systematische Lösung für die Klasse der HF-Verstärker vorgeschlagen, um das lineare und nichtlineare Verhalten zu verifizieren. Sie überträgt die wichtigsten Konzepte der digitalen Funktions-Abdeckung auf den Kontext von SystemC AMS im Besonderen und Simulationen auf Systemebene im Allgemeinen. Um eine AMS-gesteuerte Verifikation der Funktions-Abdeckung zu ermöglichen, werden zwei Parameter zur Verfeinerung der Abdeckung auf der Eingangs- und Ausgangsseite eingeführt, um systematisch Eingangsstimuli zu erzeugen und Spezifikationen zu erfassen. Das Herzstück des Ansatzes ist die Abdeckungsanalyse, die die funktionale Abdeckung des DUV misst und dem Verifikationsingenieur ein klares Feedback gibt, um die Abdeckung abzuschließen. Die Parameter zur Verfeinerung der Abdeckung müssen jedoch manuell angepasst werden, was bei komplexen Systemen und instabilem Verhalten einen Engpass darstellt.

In diesem Zusammenhang wird ein *leichtgewichtiger Ansatz zur abdeckungsgesteuerten Stimulierung* (engl. *Lightweight Coverage-Directed Stimuli Generation*, LCDG) in Betracht gezogen. CDG ist eine Verifikationsmethodik, die darauf abzielt, die überdeckung

automatisch zu erreichen, indem überdeckungsdaten und mathematische Funktionen verwendet werden, um die nächsten Iterationen der Teststimuliererzeugung zu steuern. Das Herzstück des vorgeschlagenen LCDG-Ansatzes ist eine Überdeckungsanalyse, die funktionale überdeckungsdaten von Eingabe, Ausgabe und Prüfern in Kombination mit der *Euklidischen Distanz* nutzt, um eine Überdeckungsschließung zu erreichen. Die *Euklidische Distanz* ist im Gegensatz zu *Bayesschen Netzen* oder komplexen Wahrscheinlichkeitsverteilungsfunktionen wesentlich einfacher. Im Falle von Überdeckungslücken passt die Analyse automatisch die Parameter der Überdeckungsverfeinerung an, um die Überdeckung des DUV zu erhöhen. Infolgedessen gewährleisten diese leichtgewichtigen und systematischen Ansätze eine effiziente Konvergenz und eine gründliche Verifizierung der VP.

Beitragsbereich 4 - Digitale frühe Sicherheitsvalidierung Der letzte Beitrag dieser Arbeit ist die frühe Sicherheitsvalidierung des funktional verifizierten VPs. Die digitale Umgebung für die frühe Sicherheitsvalidierung ist in Abb. 5 rosa unterlegt dargestellt. Sie besteht aus drei Hauptkomponenten: (1) Sicherheitseigenschaften, (2) Trace-Logs und (3) Kombination aus statischer und dynamischer Sicherheitsanalyse. Unter Nutzung dieser Komponenten wird in dieser Arbeit ein neuartiger Ansatz zur SOC-Sicherheitsvalidierung auf Systemebene unter Verwendung von VPs vorgeschlagen. Das Herzstück des Ansatzes ist eine skalierbare statische Informationsflussanalyse, die potenzielle Sicherheitsverletzungen wie Datenlecks und nicht vertrauenswürdigen Zugriffe, d.h. *Vertraulichkeits-* bzw. *Integritätsprobleme*, erkennen kann.

Darüber hinaus ergänzt diese Arbeit den statischen Ansatz, indem sie eine dynamische Informationsflussanalyse für VPs vorstellt. Sie befasst sich insbesondere mit dem Sicherheitsmerkmal der IP-Isolierung, das heutzutage weit verbreitet ist, z.B. werden gesicherte Memory Mapped IOs (MMIOs) oder gesicherte Adressbereiche im Falle von Speichern als nicht zugänglich markiert. Eine Möglichkeit zur Gewährleistung der Sicherheit, ist die Definition von Isolation als Informationsflusspolitik in HW unter Verwendung des Begriffs der Nichteinmischung zu definieren. Der vorgeschlagene Ansatz ermöglicht die Validierung Laufzeitverhalten eines gegebenen SOC, der mit VPs implementiert wurde, gegen Sicherheitsbedrohungsmodelle, wie z.B. Informationslecks (*Vertraulichkeit*) und unbefugtem Zugriff auf Daten in einem Speicher (*Integrität*).

Fazit Zusammenfassend lässt sich sagen, dass diese Beiträge einen hochqualitativen VP-basierten Verifizierungsablauf vorschlagen, wie von den ausfuhrlichen Experimenten der Dissertation belegt wird [Ha21]. Einer der Hauptvorteile ist die drastisch verbesserte Verifikationsqualität in Kombination mit einem deutlich geringeren Verifikationsaufwand. Einerseits reduziert dies die Anzahl der unentdeckten Fehler und erhöht die Gesamtqualität des AMS SOC. Des Weiteren wird eine qualitativ hochwertige VP-Testsuite erstellt, die für die Verifikation der unteren Abstraktionsebenen verwendet werden kann.

Literaturverzeichnis

[Ba10] Barnasconi, Martin: SystemC AMS extensions: Solving the need for speed. 2010.

[Ba13] Barnasconi, Martin; Einwich, Karsten; Grimm, Christoph; Maehne, Torsten; Vachoux, Alain et al.: Standard SystemC AMS extensions 2.0 language reference manual. Accellera Systems Initiative, 2013.

[GD10] Große, Daniel; Drechsler, Rolf: Quality-Driven SystemC Design. Springer, 2010.

[Ha17] Hassan, Muhammad; Herdt, Vladimir; Le, Hoang M.; Große, Daniel; Drechsler, Rolf: Early SoC Security Validation by VP-based Static Information Flow Analysis. In: International Conference on Computer-Aided Design. S. 400–407, 2017.

[Ha18] Hassan, Muhammad; Große, Daniel; Le, Hoang M.; Vörtler, Thilo; Einwich, Karsten; Drechsler, Rolf: Testbench Qualification for SystemC-AMS Timed Data Flow Models. In: Design, Automation and Test in Europe. S. 857–860, 2018.

[Ha19] Hassan, Muhammad; Große, Daniel; Vörtler, Thilo; Einwich, Karsten; Drechsler, Rolf: Functional Coverage-Driven Characterization of RF Amplifiers. In: Forum on Specification and Design Languages. S. 1–8, 2019.

[Ha21] Hassan, Muhammad: Enhanced Modern Virtual Prototype based Verification Flow for Heterogeneous Systems. Dissertation, University of Bremen, 2021.

[He16] Herdt, Vladimir; Le, Hoang M.; Große, Daniel; Drechsler, Rolf: Compiled Symbolic Simulation for SystemC. In: International Conference on Computer-Aided Design. S. 52:1–52:8, 2016.

[HGD20] Herdt, Vladimir; Große, Daniel; Drechsler, Rolf: Enhanced Virtual Prototyping: Featuring RISC-V Case Studies. Springer, 2020.

[HGD21a] Hassan, Muhammad; Große, Daniel; Drechsler, Rolf: System-Level Verification of Linear and Non-Linear Behaviors of RF Amplifiers using Metamorphic Relations. In: ASP Design Automation Conf. 2021.

[HGD21b] Hassan, Muhammad; Große, Daniel; Drechsler, Rolf: System Level verification of Phase-Locked Loop using Metamorphic Relations. In: Design, Automation and Test in Europe. 2021.

[In06] Initiative, Open SystemC et al.: IEEE standard SystemC language reference manual. IEEE Computer Society, S. 1666–2005, 2006.

[Le16] Le, Hoang M.; Herdt, Vladimir; Große, Daniel; Drechsler, Rolf: Towards Formal Verification of Real-World SystemC TLM Peripheral Models – A Case Study. In: Design, Automation and Test in Europe. S. 1160–1163, 2016.

Muhammad Hassan erhielt 2015 den M.Sc. in Nachrichtentechnik von der RWTH Aachen, Deutschland. Danach begann er als Doktorand in der Arbeitsgruppe Rechnerarchitektur unter der Betreuung von Prof. Rolf Drechsler. Seit 2017 ist er als Wissenschaftlicher Mitarbeiter bei dem Deutschen Forschungszentrum fur Künstliche Intelligenz (DFKI) GmbH tätig. Im Jahr 2021 erhielt er den Dr.-Ing. Titel in Informatik von der Universität Bremen. Seine aktuellen Forschungsinteressen umfassen Virtual Prototyping sowie Verifikations- und Analysetechniken mit einem besonderen Fokus auf heterogene Systeme. In diesen Bereichen veröffentlichte er mehr als 10 peer-reviewed Journal- und Konferenzbeitrage mit zwei Best-Paper-Candidates und einem Best-Paper-Award bei der DVCON Europe.

Drohnennetzwerke zur Suche und Rettung[1]

Samira Hayat[2]

Abstract: Diese Arbeit befasst sich mit dem komplexen Problem des Entwurfs von Drohnennetzwerken. Drohnenanwendungen sind sehr vielfältig und können mit traditionellen Netzwerkdesignmethoden nicht erfolgreich bewältigt werden. Alternativ schlagen wir vor, die grundlegenden Fragen für Drohnennetzwerkanwendungen zu beantworten: Welche Daten müssen übertragen werden? Wie werden die Daten von Punkt A zu Punkt B im Netzwerk übertragen? Wann (zu welchen Zeitpunkten) müssen die Daten übertragen werden? Wir stellen fest, dass Kommunikation in einem Drohnennetzwerk sowohl dem Missionsziel (für die Übertragung von Sensordaten) als auch der Missionsdurchführung (Missionskoordination) dient, weshalb wir eine abstimmbare und modulare Systemarchitektur vorschlagen. Abhängig von den Missionsanforderungen können verschiedene Kommunikations- und Koordinationsmodule hinzugefügt werden, ohne dass das gesamte System geändert werden muss. Die Systemarchitektur wird für einen Such- und Rettungseinsatz implementiert, wobei bestehende Kommunikationstechnologien im Experiment getestet und neuartige Koordinationsalgorithmen vorgeschlagen werden.

1 Einleitung

Drohnennetzwerke sind sehr vielseitig einsetzbar Hayat et al. [HYM16]. Jeder Drohneneinsatz unterliegt unterschiedlichen Anforderungen und ist abhängig von der Anzahl und dem Typ der Fluggeräte, der Größe des Einsatzgebiets, der Nutzlast und der Flugzeit, dem Datenverkehr und dem Grad an Autonomie. Um die unterschiedlichen Einsatzanforderungen für Drohnenanwendungen zu veranschaulichen, verwenden wir beispielhaft verschiedene Drohnenanwendungen.

1. Gebietskartierung erfordert die Sensorabdeckung eines bestimmten Gebiets, z. B. eines landwirtschaftlich genutzten Feldes, mit nicht mehr als zehn Drohnen, die ferngesteuert sein können. Sie sammeln Sensordaten während des Flugs und übertragen sie nach Abschluss der Mission zur Analyse an eine Bodeneinheit; diese Art der Datenübertragung ist verzögerungstolerant.
2. Bei der Netzbereitstellung besteht das Ziel der Mission darin, dem Bodenpersonal Netzkonnektivität unter unterschiedlichen Bedingungen bereitzustellen. Eine einzelne autonom fliegende HALE-Drohne (High Altitude, Long Endurance) kann eine Netzabdeckung von mehreren Kilometern gewährleisten. Alternativ können mehrere Drohnen Messwerte von Sensoren am Boden sammeln.

[1] Englischer Titel der Dissertation: “Drone Networks for Search and Rescue”

[2] Lakeside Labs GmbH, B04b, Lakeside Park, 9020, Klagenfurt, Österreich. hayat@lakeside-labs.com

3. In Katastrophensituationen, wie bei Such- und Rettungseinsätzen (Search and Rescue — SAR), ist das zu erfassende Gebiet kilometergroß. Bei manchen Missionen müssen die Drohnen dynamisch planen und autonom fliegen. Für solche Fälle wird man möglicherweise Drohnen brauchen, die sowohl Sensordaten sammeln als auch in Echtzeit an die Bodenstation übermitteln. Abhängig von der Größe des zu überwachenden Gebiets und von der Zeit, bis zu welcher eine Rückmeldung erfolgen muss, können mehrere Dutzend Drohnen nötig sein.

Wir kommen zu dem Schluss, dass das Drohnennetzwerk als auftragszentriert behandelt werden muss, in dem eine oder mehrere Drohnen eine Reihe von Aufgaben in einem bestimmten Gebiet mit einem bestimmten Grad an Autonomie ausführen. Die Missionsanforderungen haben direkten Einfluss auf das Netzwerkdesign. Die Anzahl der Drohnen im Netzwerk, das Erkundungsgebiet und die Aufgaben sowie der Grad an Autonomie variieren je nach Mission, und das gilt auch für das zugrundeliegende Netzwerkdesign.

Die oben genannten Anwendungen zeigen einen weiteren wichtigen Aspekt: In einem Drohnennetzwerk spielt die Kommunikation unterschiedliche Rollen. Sie hilft einem Team von Drohnen bei der Koordinierung und der gemeinsamen Durchführung von Missionsaufgaben, d. h., die Kommunikation ermöglicht die Koordinierung mehrerer Drohnen. Die Kommunikation ist auch ein Missionsziel, beispielsweise in den oben genannten Fällen 2 und 3. Bei allen Drohnenanwendungen wird die hohe 3D-Mobilität der Drohnen genutzt, um Daten zu sammeln, zu übertragen und zu verbreiten. Diese Übertragung kann in Echtzeit oder periodisch erfolgen oder verzögerungstolerant sein.

Um ein Drohnennetzwerk zu konzipieren, müssen wir daher die Anforderungen bestimmter Anwendungen an Kommunikation kennen und ihre Rolle in der jeweiligen Anwendung richtig einschätzen. Der herkömmliche Ansatz, die Schichten des Protokollstapels zu verbessern, kann für einige Anwendungen genügen, ist aber für andere nicht ausreichend Bashir; Mohamad Yusof [BM19]. Wir schlagen daher einen alternativen Ansatz für das Design von Drohnenkommunikationsnetzwerken vor, welcher auf den Anforderungen der Mission basiert, wobei uns SAR als implementierter Anwendungsfall dient.

2 Demonstrationen

Für den von uns vorgeschlagenen Ansatz, der als missionsorientiertes Netzwerkdesign bezeichnet wird, ergibt sich die Lösung aus der Beantwortung dreier grundlegender Fragen zur Datenübertragung:

1. Welche Daten müssen übertragen werden?
2. Wie sollten Daten zwischen den Netzwerkeinheiten übertragen werden?
3. Wann sollten Daten übertragen werden?

Um 1 zu beantworten, untersuchen wir die *missionsorientierten Kommunikationsanforderungen* von Drohnenanwendungsdomänen. Diese Domänen umfassen Anwendungen, die ähnliche Missionsziele verfolgen. Zur Beantwortung von Frage 2 untersuchen wir *experimentell die vorhandenen Kommunikationstechnologien*, um ihre Stärken und Schwächen bei der Anwendung in Drohnennetzwerken zu ermitteln. Dies vermeidet Neuerfindungen und zeigt, wieviel Aufwand für die Anpassung der Technologien an die Anforderungen von Drohnenanwendungen nötig ist. Schließlich wird zur Beantwortung von Frage 3 ein neuartiger, *mehrdimensionaler Algorithmus zur (Neu-)Planung von Pfaden* für Multidrohnensysteme vorgeschlagen. Die sich daraus ergebenden Lösungen bilden die drei Teile der vorliegenden Doktorarbeit.

2.1 Literaturverzeichnis

Eine detaillierte Literaturübersicht zu den drei Teilen der Arbeit findet sich in Hayat [Ha21]. Im Folgenden wird nur die relevanteste Literatur vorgestellt.

Missionsorientierte Kommunikationsanforderungen Der Großteil der Literatur zum Design von Drohnennetzwerken geht von verallgemeinernden Annahmen über Kommunikationsanforderungen aus Yang et al. [Ya18]. So hat beispielsweise das Third Generation Partnership Project (3GPP) die grundlegenden Anforderungen an Drohnennetzwerke für Uplink und Downlink in Bezug auf Datenraten, Zuverlässigkeit und Latenzzeit festgelegt, die als Richtlinie für die Integration von Drohnen in zukünftige Mobilfunknetze dienen Zeng et al. [ZLZ19]. Dennoch kommen viele Artikel zu dem Schluss, dass die Gestaltung von Luftverkehrsnetzen missionsspezifisch sein muss; die Herausforderungen für das Netz hängen in hohem Maße von der Rolle der Drohne im Netz ab (Netznutzer*innen versus Netzbetreiber*innen) Mozaffari et al. [Mo19]. Die Auswirkungen der Missionsziele auf die Gestaltung von Drohnennetzwerken werden auch in Shah; Kim [SK14] hervorgehoben. Im Rahmen dieser Arbeit haben wir eine umfassende Untersuchung durchgeführt, um die missionsspezifischen qualitativen und quantitativen Kommunikationsanforderungen zu ermitteln.

Experimentelle Untersuchung bestehender Drahtlostechnologien Ein umfassendes Tutorial zur analytischen, simulativen und experimentellen Bewertung der drahtlosen Kommunikation in Drohnen wird in Vinogradov et al. [Vi18] vorgestellt. Es wird gezeigt, dass die drei Ansätze einander ergänzen, da jeder gewissen Einschränkungen unterliegt. Obwohl Simulationen der beliebteste Ansatz zur Netzwerkanalyse sind Zhang et al. [ZZZ19], stützen sie sich stark auf Kanalmodelle. Die Kanalmodellierung für Drohnennetzwerke ist aufgrund der hohen Mobilität der Drohnen und der daraus resultierenden hochdynamischen Funkbedingungen sehr mühsam und daher Experimente in der realen Welt erforderlich machen. Es gibt keine formale Neuzuweisung von Frequenzen speziell für Netze in der Luft Marcus [Ma14]; Experimente haben sich hauptsächlich auf Technologien im unlizenzierten Spektrum konzentriert Asadpour et al. [As13]. Die Nutzung von Drohnen im Zusammenhang mit zellularen Netzen wie Long Term Evolution (LTE) ist relativ neu Zhang

et al. [ZZZ19]. Unsere Arbeit zeigt eine bessere Performanz sowohl im unlizenzierten (vgl. Asadpour et al. [As13] als auch im lizenzierten Spektrum (vgl. Marques et al. [Ma19]).

Mehrdimensionale Pfadplanung Die meisten Forschungsarbeiten zur Zielsuche mit mehreren Drohnen berücksichtigen eine scheibenförmige Konnektivität zwischen den Drohnen (siehe [YP12]). Ein solch restriktiver Ansatz (eingeschränkte Konnektivität zur Bodenstation) — sei es kontinuierliche [La19] oder periodische Konnektivität [Ce15] — führt zu einer Verschlechterung der Abdeckungsleistung. Für unseren Pfadplanungsansatz behandeln wir Konnektivität als Ziel und nicht als Einschränkung. Im Vergleich zu den oben genannten Arbeiten ermöglicht unser Ansatz eine flexible Pfadplanung in Bezug auf die Parameter Abdeckung und Konnektivität. Basierend auf den Anforderungen der Mission erlaubt der Algorithmus einen Kompromiss zwischen den beiden Parametern. Die Arbeit [Fl13] beschreibt eine Lösung, die unserer am ehesten entspricht. Die Autoren schlagen einen Ansatz der Mixed-Integer Linear Programming (MILP) vor, bei dem die Drohnen nur als Relais fungieren. In [Ma17] wird ein MANET-Algorithmus vorgestellt, der optimale Durchsatzpfade gewährleistet, aber spezielle Relaisknoten erfordert, die nicht an der Suche beteiligt sind. In ähnlicher Weise werden Drohnen zur Optimierung der Dienstgüte (Quality-of-service — QoS; Netzwerkkonnektivität und Durchsatz) als Basisstationen in der Luft in [LZZ19] eingesetzt. Die Zuweisung spezieller Aufgaben an eine Reihe von Drohnen im Netz kann zu einer ineffizienten Ressourcennutzung führen. Wir nutzen daher alle Drohnen für die Abdeckung. Kommunikationsaufgaben (Relaying/Datenübermittlung) werden den Drohnen zugewiesen, wenn eine Informationsübertragung erforderlich ist. Der Algorithmus stellt die bestmöglichen QoS-Pfade am Ende der Mission sicher.

2.2 Kommunikationsanforderungen

Wir haben eine umfassende Untersuchung von Hayat et al. [HYM16] durchgeführt, um die Kommunikationsanforderungen in Drohnennetzwerken systematisch zu ermitteln. Wir kommen zu dem Schluss, dass diese Anforderungen abhängig sind von: (i) Anwendungen und (ii) Systemdesign. Für die SAR-Anwendung wird an die Kommunikation eine größere Anforderung gestellt als für die Gebietskartierung. Das Systemdesign beeinflusst auch die im Netzwerk übertragenen Daten; unterschiedliche Teamkoordinationsmethoden (zentralisiert oder verteilt) wirken sich auf das zugrundeliegende Netzwerkdesign sowie auf die Menge der übermittelten Daten aus. Wir haben die minimalen quantitativen Datenübertragungsanforderungen für verschiedene Anwendungsbereiche sowie die qualitativen Anforderungen der verschiedenen Systemimplementierungen dargestellt und sind zu dem Schluss gekommen, dass keine einzelne Kommunikationslösung die Anforderungen der verschiedenen Drohnenanwendungen und der verschiedenen Koordinationslösungen erfüllen kann. Dies motivierte die Entwicklung einer Proof-of-Concept-Architektur. Die vorgeschlagene Architektur behandelt Kommunikation, Koordination, Sensorik und die Drohnenhardware als Systemmodule mit klar definierten Interaktionen zwischen den Modulen. Eine solche modulare Architektur würde es den Systementwickler*innenn ermöglichen, ein einzelnes

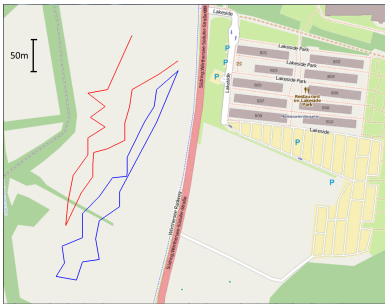
Modul entsprechend den Missionsanforderungen zu modifizieren, ohne dass ein komplettes System neu entwickelt werden müsste. Wir haben das System für SAR getestet.

2.3 Kommunikationsexperimente

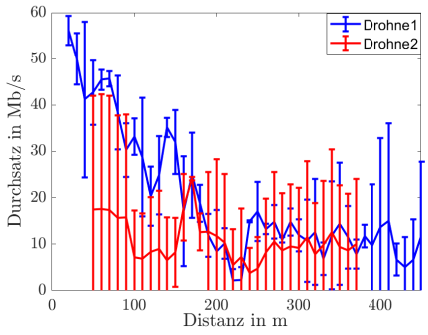
Mit genau definierten System- und Missions-QoS-Anforderungen im Hinterkopf haben wir bestehende mobile Netzwerktechnologien mit Drohnen evaluiert. Die Testtechnologien stammen sowohl aus dem unlizenzierten (IEEE 802.11a/n/ac) als auch aus dem lizenzierten (LTE-Advanced oder LTE-A) Spektrum und haben keine drohnenspezifischen Verbesserungen erfahren. Eine solche Bewertung anhand von Experimenten führt zu wertvollen Schlussfolgerungen. Wir können eine Neuerfindung vermeiden, wenn die bestehenden Technologien den Anforderungen bestimmter Drohnenanwendungen genügen, oder wir können die Mängel der aktuellen Lösungen erkennen und – wo erforderlich – unser Augenmerk auf Änderungen und Verbesserungen legen.

Unlizenziertes Spektrum

Wir haben eine Reihe von Experimenten mit 802.11a/n/ac durchgeführt. Eine Drohne flog auf einem geraden Weg von einer Basisstation (BS) weg und übertrug Daten entweder direkt oder über eine Relais-Drohne an die BS. Wir analysierten sowohl Zugangspunkte (AP) als auch Mesh-Netzwerke zur Reichweitenerweiterung. Die Experimente lieferten umfassende Einblicke in die reale Leistung der unlizenzierten Technologien. Da der Zweck der Experimente darin bestand, zu verstehen, wie die Technologien für SAR abschneiden könnten, stellen wir in dieser Zusammenfassung die Ergebnisse für ein Ad-hoc-Szenario mit mehreren Drohnen vor, bei dem zwei Drohnen (Drohne1 und Drohne2) ein Gebiet abdecken und die erfassten Daten an eine BS senden (siehe Abbildung 1).



(a) Flugpwege der beiden Drohnen (blaue und rote durchgezogene Linien).



(b) Durchsatz.

Abb. 1: Abdeckungsszenario mit zwei Drohnen (Drohne1 und Drohne2), die Luft-Boden-Verkehr im Ad-hoc-Modus mit 802.11n senden.

In mehreren Durchläufen beanspruchte Drohne1, welche näher an der BS war, zu Beginn des Tests einen größeren Teil des Kanals: Der durchschnittliche Durchsatz von Drohne1 war

Höhe	RSSNR Korrelation	SIR Korrelation
10 m	0.40	0.31
50 m	0.58	0.27
100 m	0.59	0.36
150 m	0.78	0.63

Tab. 1: Kreuzkorrelation des Durchsatzes mit dem Signal-Rausch- (RSSNR) und Signal-Störungs-Verhältnis (SIR) (Durchschnittswerte über vier Messreihen)

mit 45 Mb/s mehr als doppelt so hoch wie der von Drohne2. Im weiteren Verlauf des Tests näherten sich die Abstände der Drohnen zum BS immer mehr an: Bei 100 s waren beide Drohnen 200 m von der BS entfernt. Der durchschnittliche Durchsatz der beiden Geräte glich sich nach 200 m an (Abbildung 1). Das Experiment wurde beendet, als Drohne1 450 m erreicht hatte.

Das Experiment zeigt, dass 802.11n über kurze Entfernungen einen beträchtlichen Durchsatz bieten kann (im Durchschnitt mehr als 10 Mb/s für beide Sender). Dieser Wert ist zufriedenstellend, wenn man die durchschnittlichen Durchsatzanforderungen des Großteils des Echtzeitverkehrs berücksichtigt. Der momentane Durchsatz schwankt aufgrund der Mobilität erheblich. Die Netzfairness scheint in einem realen Übertragungsszenario mit mobilen Sendern zu gelten.

Lizenziertes Spektrum

Die Technologien im unlizenzierten Spektrum sind dann von Vorteil, wenn keine Netzinfrastruktur vorhanden ist. Für Anwendungen mit Drohnenschwärmen oder für die Zustellung per Drohne sind sie nicht geeignet. Im ersten Fall können die Drohnen im Schwarm viel höhere Anforderungen an die Kommunikation stellen (Durchsatz, Zuverlässigkeit, Fairness usw.), als sie mit Wi-Fi erfüllt werden können. Im zweiten Fall muss eine Infrastruktur vorhanden sein, um die Mobilität der Drohnen über sehr große Entfernungen zu verfolgen. Um die Sicherheit des Drohnennetzes zu gewährleisten, wurden lizenzierte Technologien für die Kommunikation genutzt. Bei den Experimenten mit lizenzierten Frequenzen haben wir LTE-A verwendet. Solche Experimente sind für die Entwickler*innen von Drohnennetzen interessant, da die Mobilfunknetze für terrestrische Nutzer*innen optimiert sind und die Hauptkeule der Antennen nach unten gerichtet ist.

Die Experimente zeigten, dass der durchschnittliche Durchsatz mit der Höhe abnimmt und eine starke Kreuzkorrelation zwischen dem momentanen Durchsatz und der RSSNR besteht, die mit der Höhe zunimmt (siehe Tabelle 1). In großen Höhen (150 m) ist der Durchsatz auch mit dem SIR korreliert. Die Anzahl der Basisstationen (eNBs in LTE-A) mit Sichtverbindung zu einer Drohne nimmt mit der Höhe zu [BCP16]. Die Störsignale dieser eNBs werden von der Drohne mit einem Signalleistung (RSRP) empfangen, das dem des beabsichtigten Signals ähnelt, was wiederum den Signal-Störungs-Verhältnis (SIR)-Wert senkt.

Eine weitere Beobachtung in großen Höhen ist, dass die Drohnen von eNB-Antennen-Nebenkeulen bedient werden, die über eine enge Winkelabdeckung verfügen; in größeren Höhen erleben die Drohnen sehr häufige Handover, was ebenfalls zum Durchsatzrückgang beiträgt. Zelluläre Netze sind eine für Drohnenanwendungen wünschenswerte Technologie, jedoch gibt es was die Integration von Drohnen betrifft noch offene Fragen wie Antennenneigung und Strahlformung.

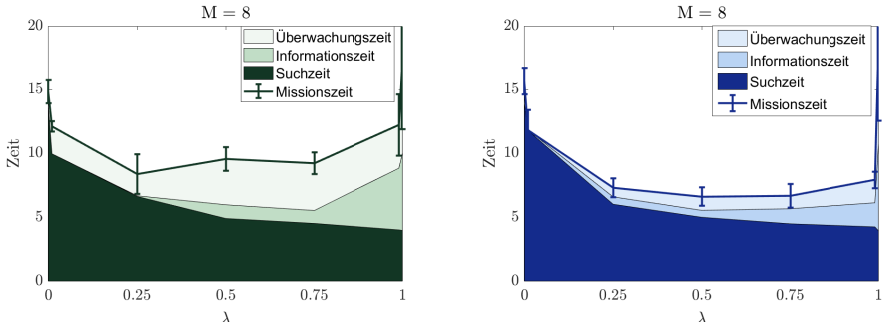
2.4 Mehrdimensionale Pfadplanung

Drohnennetzwerke unterscheiden sich von anderen in der einzigartigen Mobilitätscharakteristik von Drohnen. Die im Rahmen dieser Arbeit durchgeführten Experimente haben gezeigt, dass die bestehenden Kommunikationstechnologien nicht in der Lage sind, die hohe Mobilität von Drohnen zu bewältigen. Das Routing von Daten durch ein solches Netzwerk ist eine Herausforderung, insbesondere, wenn bestimmte Prioritäten bei der Datenübertragung (periodisch, in Echtzeit) erfüllt werden müssen. Wir haben einen Pfadplanungsalgorithmus für die Anwendungsschicht entwickelt, der die hohe Mobilität von Drohnen für das Datenrouting in einem Drohnennetzwerk nutzt und die Frage beantwortet, *wann* Daten übertragen werden sollen. Wir haben festgestellt, dass die meisten Drohnenanwendungen Abdeckung und/oder Konnektivität als Missionsziele haben. Unser Algorithmusentwurf zielt sowohl auf Abdeckung als auch auf Konnektivität. Dieser Mehrzielalgorithmus ist mit einem Parameter λ abstimmbar, d.h. ein Missionsziel kann je nach den Anforderungen der Mission mit folgender Gleichung priorisiert werden:

$$\tau = \lambda\tau_{cov} + (1 - \lambda)\tau_{conn} \tag{1}$$

Wobei τ die Gesamtzielfunktion darstellt, τ_{cov} die Abdeckungsfunktion und τ_{conn} die Konnektivitätsfunktion. Wenn die Abdeckung eine höhere Priorität hat als die Konnektivität gilt $\lambda > 0,5$ und umgekehrt.

Mit Blick auf die SAR-Mission haben wir dann die Aufgaben der Mission identifiziert. Diese sind: (i) Suchaufgabe, um ein Ziel zu lokalisieren, (ii) Informationsaufgabe, um die BS über den Standort des Ziels zu informieren, und (iii) Überwachungsaufgabe, um eine Relaisverbindung zwischen dem Ziel und der BS herzustellen, um das Ziel kontinuierlich zu überwachen. Die Suchaufgabe sorgt für die Gebietsabdeckung, während die Informations- und Überwachungsaufgaben die Netzwerkkonnektivität ermöglichen. Der Algorithmus nutzt die Netzwerkressourcen optimal aus, d. h. alle Drohnen beteiligen sich an der Ausführung aller verfügbaren Aufgaben, und die wichtigsten Aufgaben werden zuerst ausgeführt. SAR-Missionen sind zeitkritisch, ein schnellerer Abschluss ist wünschenswert. Daher steht τ in Gleichung 1 für die Missionszeit, τ_{cov} für die Zeit zur Erfüllung der Aufgaben der Abdeckung und τ_{conn} für die Zeit zur Erfüllung der konnektivitätsbezogenen Aufgaben. Genetic Algorithm (GA) Optimierung um Gleichung 1 zu minimieren.



(a) SIC+ (b) SICQ

Abb. 2: Zeit versus λ für (a) SIC+ und (b) SICQ. SAR mit acht Drohnen ($M = 8$). Zeit in Schritten.

Wir haben dann mehrere Strategien vorgeschlagen, die auf der Datenübermittlung (inform-first; IF), der Weiterleitung (connect-first; CF) und einer Hybridstrategie (simultaneous-inform-and-connect; SIC+) basieren. Die Hybridstrategie übertraf die beiden anderen Strategien. Wir haben auch gezeigt, dass die Gesamtzeit für den Abschluss der Mission im Vergleich zu Pfaden ohne QoS (SIC+) reduziert wurde, wenn die Drohnenpfade QoS (SICQ) gewährleisteten. Dies ist in Abbildung 2 zu sehen. Mit SICQ wird die Konnektivität verbessert, ohne dass die Abdeckung im Vergleich zu SIC+ stark beeinträchtigt wird.

3 Schlussbemerkungen

Die Fokussierung auf das *Was*, *Wie* und *Wann* von Drohnennetzwerken soll Entwickler*innen von Multidrohnennetzwerken und -systemen helfen

- die Kommunikationsanforderungen von Drohnennetzwerken mit Hilfe von Anwendungsbeispielen und Implementierungslösungen zu verstehen,
- die vorgeschlagene modulare Systemarchitektur zu nutzen, um ihre eigenen Kommunikations- und Koordinationslösungen zu testen, ohne dass ein komplettes System-Redesign erforderlich ist,
- die Vorteile und Schwächen der bestehenden Kommunikationstechnologien im lizenzierten und unlizenzierten Spektrum bei der Anwendung auf Drohnennetze zu verstehen und zu erkennen und
- den abstimmbaren Pfadplanungsalgorithmus und die Strategien der Anwendungsschicht zu nutzen, um bestimmte Garantien in Bezug auf die Dienstgüte und die Missionserfüllungszeiten unter den gegebenen Bedingungen (Gebiet, Technologie und Drohnendichte) zu gewährleisten.

Obwohl SAR als Anwendungsfall gewählt wurde, kann unser Ansatz den Systementwickler*innen für jede beliebige Multi-Drohnen-Mission von Nutzen sein.

Literatur

[As13] Asadpour, M.; Giustiniano, D.; Hummel, K. A.; Heimlicher, S.: Characterizing 802.11n Aerial Communication. In: Proc. ACM MobiHoc Workshop on Airborne Networks and Communications. 2013.

[BCP16] Bergh, B. V. D.; Chiumento, A.; Pollin, S.: LTE in the Sky: Trading off Propagation Benefits with Interference Costs for Aerial Nodes. IEEE Communications Magazine/, 2016.

[BM19] Bashir, M. N.; Mohamad Yusof, K.: Green Mesh Network of UAVs: A Survey of Energy Efficient Protocols across Physical, Data Link and Network Layers. In: Proc. Intl. Conf. on Big Data and Smart City (ICBDSC). 2019.

[Ce15] Cesare, K.; Skeele, R.; Yoo, S.; Zhang, Y.; Hollinger, G.: Multi-UAV exploration with limited communication and battery. In: Proc. IEEE Intl. Conf. on Robotics and Automation (ICRA). 2015.

[Fl13] Flushing, E. F.; Kudelski, M.; Gambardella, L. M.; Caro, G. A. D.: Connectivity-aware planning of search and rescue missions. In: Proc. IEEE Intl. Symp. on Safety, Security, and Rescue Robotics (SSRR). 2013.

[Ha21] Hayat, S.: Drohnennetze für Suche und Rettung, Dissertation, Klagenfurt University, 2021.

[HYM16] Hayat, S.; Yanmaz, E.; Muzaffar, R.: Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint. IEEE Communications Surveys & Tutorials 18/, 2016.

[La19] Lamine, A.; Mguis, F.; Snoussi, H.; Ghedira, K.: Coverage Optimization using Multiple Unmanned Aerial Vehicles with Connectivity Constraint. In: Proc. Intl. Wireless Communications Mobile Computing Conf. (IWCMC). 2019.

[LZZ19] Liu, L.; Zhang, S.; Zhang, R.: CoMP in the Sky: UAV Placement and Movement Optimization for Multi-User Communications. IEEE Transactions on Communications 67/, 2019.

[Ma14] Marcus, M.: Spectrum policy challenges of UAV/drones [Spectrum Policy and Regulatory Issues]. IEEE Wireless Communications 21/, 2014.

[Ma17] Magán-Carrión, R.; Camacho, J.; García-Teodoro, P.; Feo Flushing, E.; Di Caro, G.: A Dynamical Relay node placement solution for MANETs. Computer Communications 114/, 2017.

[Ma19] Marques, H.; Marques, P.; Ribeiro, J.; Alves, T.; Pereira, L.: Experimental Evaluation of Cellular Networks for UAV Operation and Services. In: Proc. IEEE 24th Intl. Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD). 2019.

[Mo19] Mozaffari, M.; Saad, W.; Bennis, M.; Nam, Y.; Debbah, M.: A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems. IEEE Communications Surveys Tutorials 21/, 2019.

[SK14] Shah, B.; Kim, K.: A survey on three-dimensional wireless ad hoc and sensor networks. Intl. Journal of Distributed Sensor Networks 10/, 2014.

[Vi18] Vinogradov, E.; Sallouha, H.; Bast, S. D.; Azari, M. M.; Pollin, S.: Tutorial on UAVs: A Blue Sky View onWireless Communication. Journal of Mobile Multimedia 14/, 2018.

[Ya18] Yang, G.; Lin, X.; Li, Y.; Cui, H.; Xu, M.; Wu, D.; Rydén, H.; Redhwan, S. B.: A Telecom Perspective on the Internet of Drones: From LTE-Advanced to 5G, 2018.

[YP12] York, G.; Pack, D. J.: Ground Target Detection Using Cooperative Unmanned Aerial Systems. Journal of Intelligent and Robotic Systems 65/, 2012.

[ZLZ19] Zeng, Y.; Lyu, J.; Zhang, R.: Cellular-Connected UAV: Potentials, Challenges and Promising Technologies. IEEE Wireless Communications 26/, 2019.

[ZZZ19] Zhang, S.; Zeng, Y.; Zhang, R.: Cellular-Enabled UAV Communication: A Connectivity-Constrained Trajectory Optimization Perspective. IEEE Transactions on Communications 67/, 2019.

Samira Hayat wurde 1981 in Pakistan geboren. Im Jahr 2011 schloss sie ihr Masterstudium Telekommunikation an der Universität Trient ab (Note 109/100) und begann als wissenschaftliche Mitarbeiterin am Institut für Vernetzte und Eingebettete Systeme der Universität Klagenfurt. Dort schloss sie im Jahr 2021 ihr Doktorat der Technischen Wissenschaften mit einer Dissertation im Bereich der Multidrohnensysteme mit ausgezeichnetem Erfolg ab. Eine ihrer **Publikationen** als Erstautorin wird mittlerweile mehr als 850 Mal zitiert. Während ihres Doktoratsstudiums war sie Gastwissenschaftlerin an der Carnegie Mellon University (USA, 2016) und beschäftigte sich mit ethischen Aspekten von Drohnen. Im Zusammenhang mit ihrer Forschung hielt sie Vorträge bei TedxCERN, re:publica, Ars Electronica Festival und wurde 2018 als Change Maker zu den Goalkeepers der Gates' Foundation eingeladen. Sie arbeitet seit einigen Jahren als Senior Researcher bei der Lakeside Labs GmbH. Außerdem gründet sie ein Unternehmen, das auf ihrer Doktorarbeit basiert. Ihre Unternehmensidee wurde ausgewählt, um auf dem Europäischen Forum und in Forbes (Deutschland) präsentiert zu werden. Zudem wurde sie für die Teilnahme am Falling Walls Festival in Berlin (2018) eingeladen.

Steffen Hölldobler et al. (Hrsg.): Ausgezeichnete Informatikdissertationen 2021,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2022 91

Werkzeuge und Methoden zum Lösen von Problemen mittels Baumweite[1]

Markus Hecher[2]

Abstract: In den letzten Jahrzehnten konnte ein beachtlicher Fortschritt im Bereich der Aussagenlogik verzeichnet werden, der sich durch überwältigend schnelle Computerprogramme (Solver) zur Lösung aussagenlogischer Formeln äußert. Einer der Gründe dieser Schnelligkeit befasst sich mit strukturellen Eigenschaften von Probleminstanzen, zum Beispiel der sogenannten Baumweite, welche versucht zu messen, wie groß der Abstand von Probleminstanzen zu einfachen Strukturen (Bäumen) ist. Diese Arbeit befasst sich mit Problemen der Künstlichen Intelligenz (KI) sowie Baumweite-basierenden Methoden und Werkzeugen zum Lösen dieser. Wir präsentieren einen neuen Typ von Problemreduktion, den wir als „zerlegungsangeleitet" bezeichnen. Dieser ist die Basis, um eine lange offen gebliebene Frage betreffend quantifizierter, aussagenlogischer Formeln (QBF) bei beschränkter Baumweite zu zeigen. Die Lösung der Frage ermöglicht ein neues Meta-Werkzeug zum Beweisen präziser unterer Laufzeitschranken einer Vielzahl von Problemen der KI. Trotz dieser Schranken implementieren wir einen Solver für Erweiterungen von SAT, der Baumweite effizient ausnutzt.

1 Einleitung

In den letzten Jahrzehnten konnte ein beachtlicher Fortschritt im Bereich der *Aussagenlogik* [Bi09; KL99] verzeichnet werden. Dieser äußerte sich dadurch, dass für das wichtigste Problem in diesem Bereich, genannt „SAT", welches sich mit der Erfüllbarkeit einer gegebenen aussagenlogische Formel befasst, überwältigend schnelle Computerprogramme („Solver") entwickelt wurden. Diese Solver liefern eine beeindruckende Leistung, weil sie oft selbst Probleminstanzen mit mehreren Millionen von Variablen spielend leicht lösen können. Das ist deswegen so bemerkenswert, weil SAT einer der bekanntesten Vertreter der NP-vollständigen Probleme ist [Co71]. Vom theoretischen Standpunkt aus bedeutet dies keine gute Nachricht, da solche Probleme im Allgemeinen nicht effizient gelöst werden können (angenommen P $\neq$ NP). Mit der Zeit hat sich sogar eine stärkere Annahme als P $\neq$ NP entwickelt, die wissenschaftlich weitestgehend akzeptiert und *Exponentialzeithypothese (EZH)* [IPZ01] genannt wird. Diese Hypothese besagt, dass man im schlimmsten Fall für das Lösen einer aussagenlogischen Formel exponentielle Laufzeit in der Anzahl der Variablen benötigt. Dieser vermeintliche Widerspruch zwischen Praxis und Theorie ist noch immer nicht vollständig geklärt, denn wahrscheinlich gibt es viele ineinandergreifende Gründe für die Schnelligkeit aktueller SAT-Solver. Einer davon befasst sich mit strukturellen Eigenschaften von Probleminstanzen, die indirekt und intern von diesen Solvern ausgenützt werden, was zumindest theoretisch demonstriert wurde [AFT11].

[1] Originaltitel in englischer Sprache: "Advanced Tools and Methods for Treewidth-Based Problem Solving".

[2] TU Wien, Logic and Computation, Favoritenstraße 9–11, 1040 Wien, Österreich, hecher@dbai.tuwien.ac.at.

Die Dissertation beschäftigt sich mit solchen strukturellen Eigenschaften, nämlich mit der sogenannten *Baumweite* [RS86]. Die Baumweite ist sehr gut erforscht und versucht zu messen, wie groß der Abstand von Probleminstanzen zu Bäumen ist (Baumnähe). Allerdings ist dieser Parameter sehr generisch und bei Weitem nicht auf Problemstellungen der Aussagenlogik beschränkt. Tatsächlich gibt es viele weitere Probleme, die parametrisiert mit Baumweite in polynomieller Zeit gelöst werden können. Ebenso gibt es viele Probleme in der Wissensrepräsentation und Künstlichen Intelligenz (KI), die bei beschränkter Baumweite in polynomieller Zeit gelöst werden können, obwohl man davon ausgeht, dass sie härter sind als SAT. Ein prominentes Beispiel solcher Probleme ist QSAT, welches sich mit der Gültigkeit einer gegebenen quantifizierten, aussagenlogischen Formel (QBF) befasst. Dies sind aussagenlogische Formeln, wobei gewisse Variablen existenziell bzw. universell quantifiziert werden können [Bi09; KL99]. Ein weiterer Vertreter solcher Probleme beschäftigt sich mit der Gültigkeit eines Antwortmengen (ASP) Programms [BET11], welches zusätzlich zur Erfüllbarkeit (SAT) fordert, dass Variablen nur dann auf wahr gesetzt werden, wenn eine Begründung vorliegt. Bemerkenswerterweise wird auch im Zusammenhang mit der Baumweite, ähnlich zu Methoden der klassischen Komplexitätstheorie, die Komplexität (Härte) solcher Probleme quantifiziert, was die exakte Laufzeitabhängigkeit beim Problemlösen in der Baumweite (Stufe der Exponentialität) beschreibt.

Beitrag. Wir präsentieren Methoden und Werkzeuge, um präzise Laufzeitresultate (*obere Laufzeitschranken*) für prominente Fragmente der Antwortmengenprogrammierung (ASP), welche ein kanonisches Paradigma zum Lösen von Problemen der Wissensrepräsentation darstellt, zu erhalten. Unsere Resultate basieren auf dem Konzept der dynamischen Programmierung, die angeleitet durch eine sogenannte Baumzerlegung und ähnlich dem Prinzip „Teile-und-herrsche“ funktioniert. Solch eine *Baumzerlegung* ist eine konkrete, strukturelle Zerlegung einer Probleminstanz, die sich stark an der Baumweite orientiert.

Des Weiteren präsentieren wir einen neuen Typ von Problemreduktion, den wir als „*decomposition-guided (DG)*“, also „zerlegungsangeleitet“, bezeichnen. Dieser Reduktionstyp erlaubt es, Baumweiteerhöhungen und -verringerungen während einer Problemreduktion von einem bestimmten Problem zu einem anderen Problem präzise zu untersuchen und zu kontrollieren. Zusätzlich ist dieser neue Reduktionstyp die Basis, um eine lange offen gebliebene Frage betreffend quantifizierter, aussagenlogischer Formeln zu zeigen. Tatsächlich sind wir damit in der Lage, präzise *untere (Laufzeit-)schranken*, unter der Annahme der Exponentialzeithypothese, für das Problem QSAT bei beschränkter Baumweite zu zeigen. Mit anderen Worten können wir mit diesem Konzept der DG-Reduktionen zeigen, dass das Problem ℓ-QSAT (QSAT beschränkt auf Quantorenrang ℓ) parametrisiert mit Baumweite k im Allgemeinen nicht besser als in einer Laufzeit, die ℓ-fach exponentiell in der Baumweite und polynomiell in der Instanzgröße ist[3]. Dieses Resultat präzisiert die Beobachtung, dass das Lösen von QSAT für Baumweite hierarchische Laufzeiten verursacht [AO14; PV06], und hebt auf nicht-inkrementelle Weise ein bekanntes Ergebnis für

[3] „ℓ-fache Exponentialität“ meint eine Laufzeit eines Turms der Zahlen 2 der Höhe ℓ mit k an der Spitze.

Quantorenrang 2 [LM17] auf beliebige Quantorenränge. Damit beantworten wir via DG-Reduktionen auf konstruktive Weise eine Frage betreffend deckender unterer Schranken, die seit 2004 offen geblieben ist [Ch04]. Die Antwort darauf hat weitere Konsequenzen.

Das Resultat über die untere Schranke des Problems QSAT erlaubt es, ein *neues Meta-Werkzeug* zum Beweisen unterer Schranken vieler Probleme der Wissensrepräsentation und künstlichen Intelligenz, zu etablieren. In weiterer Konsequenz können wir damit auch zeigen, dass die oberen Schranken sowie die DG-Reduktionen dieser Arbeit unter der Hypothese EZH „eng“ sind, d.h., sie können wahrscheinlich nicht mehr signifikant verbessert werden. Die Ergebnisse betreffend der unteren Schranken für QSAT und das dazugehörige Werkzeug konstituieren eine präzise Hierarchie von über Baumweite parametrisierte Laufzeitklassen. Diese Laufzeitklassen können verwendet werden, um die Härte von KI-relevanten Probleme für das Ausnützen von Baumweite zu quantifizieren und entsprechend ihrer Baumweite-Laufzeitabhängigkeit zu kategorisieren.

Schlussendlich und trotz der genannten Resultate betreffend unterer Schranken sind wir in der Lage, eine effiziente Implementierung von Algorithmen basierend auf dynamischer Programmierung, die entlang einer Baumzerlegung angeleitet wird, zur Verfügung zu stellen. Dabei funktioniert unser Ansatz dahingehend, passende *Abstraktionen* von Instanzen zu finden, die dann sukzessive und auf rekursive Art und Weise verfeinert und verbessert werden. Inspiriert durch die enorme Effizienz und Effektivität der SAT-Solver, ist unsere Implementierung ein *hybrider Ansatz*, weil sie den starken Gebrauch von SAT-Solvern zum Lösen diverser Subprobleme, die während der dynamischen Programmierung auftreten, pflegt. Dabei stellt sich heraus, dass sich der resultierende Solver unserer Implementierung in Bezug auf die Effizienz beim Lösen von zwei kanonischen, SAT-verwandten Zählproblemen kompetitiv zu bestehenden Solvern verhält. Tatsächlich können wir Instanzen, die die oberen Schranken von Baumweite 260 übersteigen, lösen, womit die Wichtigkeit der Berücksichtigung von Baumweite in modernen Solver-Designs unterstrichen wird.

Überblick und Struktur. Als Kurzüberblick über Schlüsselresultate dieser Arbeit und einiger bestehender Resultate, stellen wir Tabelle 1 zur Verfügung, welche einen Auszug ausgewählter unterer und oberer Schranken, die in der Doktorarbeit bewiesen werden, kurz zusammenfasst. Bei den dargestellten Problemen handelt es sich um Varianten von SAT, QSAT sowie wichtiger Fragmente von ASP, die unter anderem essenziell für das Modellieren in der Wissensrepräsentation sind. Diese Tabelle wird in der Arbeit präzisiert und deutlich ergänzt, um auf die Konsequenzen für weitere Probleme der KI aufmerksam zu machen. Zu diesem Zweck verweisen wir allerdings auf auf Kapitel 6 [He21, Tabelle 6.1].

In der nächsten Sektion wird kurz auf notwendige Grundlagen eingegangen, danach wird in Sektion 3 das Konzept der „decomposition-guided“ Reduktionen erläutert, was einer vereinfachten Darstellung von Kapitel 4 der Doktorarbeit entspricht. In den folgenden Sektionen 4 und 5 stellen wir neue untere Schranken für QSAT und ASP unter Verwendung von DG-Reduktionen vor, was Kapitel 5.1 und 5.2 behandelt. Sektion 6 skizziert entspre-

Problem	Laufzeitabhängigkeit in der Baumweite k			
	Exponentialität	Laufzeit*	Obere Schranke	Untere Schranke
SAT	einfach exponentiell	$2^{\Theta(k)}$	△[SS10]	▽[IPZ01]
TIGHT ASP	einfach exponentiell	$2^{\Theta(k)}$	▲ Thm. 3.8	▼ Prop. 3.9
NORMAL ASP	übereinfach exp.	$2^{\Theta(k \cdot \log(k))}$	▲ Thm. 3.16	▼ **Thm. 3**
ι-TIGHT ASP	übereinfach exp.	$2^{\Theta(k \cdot \log(\iota))}$	▲ Thm. 4.27	▼ Corr. 4.28
ASP	doppelt exponentiell	$2^{2^{\Theta(k)}}$	△[JPW09]	▼ **Thm. 2**
ℓ-QSAT, Quantorenrang ℓ	ℓ-fach exponentiell	$\mathsf{tower}(\ell, \Theta(k))$	△[Ch04]	▼ **Thm. 1**

Tab. 1: Auszug einiger Schlüssellaufzeitresultate der Arbeit, bestehend aus oberen Schranken sowie dazu passenden Härteergebnissen (untere Schranken) für Probleme parametrisiert mit Baumweite. Ideen fett gedruckter Aussagen werden in dieser Fassung skizziert. Die Spalte „Laufzeit*“ berücksichtigt keine Faktoren, die *polynomiell* ($\mathrm{poly}(n)$) in der Instanzgröße n sind. *Die Funktion* $\mathsf{tower}(\ell, k)$ ist ein Turm von Exponenten von 2 der Höhe ℓ mit k an der Spitze. Bekannte obere Schranken sind durch „△“ markiert, während neue Ergebnisse durch „▲“ markiert sind. Untere Schranken nehmen die EZH an; neue Ergebnisse sind durch „▼“, bekannte Resultate durch „▽“ markiert.

chend Kapitel 7 einen Überblick über Ideen zum praktischen Lösen von Problemen mit Baumweite, sodass trotz theoretischer Schranken beachtliche Ergebnisse möglich sind.

2 Grundlagen

Eine *Formel* F ist eine Konjunktion von *Klauseln*, die Disjunktionen von Variablen oder deren Negation sind, vgl. [Bi09]. Der *Quantorenrang* einer *quantifizierten Bool'schen Formel (QBF)* $Q = \exists V_1 . \forall V_2 . \cdots \exists V_\ell . F$ ist die Anzahl ℓ alternierender Quantoren [KL99].

Ein *Antwortmengen (ASP) Programm* [BET11] ist eine Menge an Regeln der Form $H \leftarrow B^+, B^-$ über Variablen-Mengen H, B^+ und B^- mit der intuitiven Bedeutung, dass, wenn alle Variablen in B^+, aber keine Variablen in B^-, hergeleitet werden können, eine Variable in H hergeleitet werden muss. Ein Programm ist *normal*, wenn $|H| = 1$, und „*tight*“, wenn es keine zyklischen Abhängigkeiten über alle Regeln zwischen Variablen in B^+ und H gibt.

Für Formel, QBF oder Program $\mathcal{U}$, gibt $\mathsf{var}(\mathcal{U})$ die *Menge ihrer Variablen* an. Der *Primalgraph* $\mathcal{G}_\mathcal{U}$ von Formel, QBF oder Programm $\mathcal{U}$ hat als Knoten $\mathsf{var}(\mathcal{U})$ mit einer Kante zwischen je zwei Variablen, die in einer gemeinsamen Klausel oder Regel vorkommen.

Eine *Baumzerlegung* $\mathcal{T} = (T, \chi)$ eines Graphen $\mathcal{G}$ besteht aus Baum T und Funktion χ, die jedem Knoten t in T eine Menge an Knoten von $\mathcal{G}$ zuordnet [RS86]. Es muss gelten (i) *Knoten abgedeckt:* für jeden Knoten v von $\mathcal{G}$ gibt es einen Knoten t in T, sodass $v \in \chi(t)$; (ii) *Kanten abgedeckt:* für jede Kante e von $\mathcal{G}$ gibt es einen Knoten t in T, sodass $e \subseteq \chi(t)$; (iii) *Verbundenheit:* wenn für je drei Knoten t_1, t_2, t_3 von T, t_2 auf dem eindeutigen Pfad von t_1 nach t_3 liegt, dann $\chi(t_1) \cap \chi(t_3) \subseteq \chi(t_2)$. Die *Weite* von $\mathcal{T}$ ist der größte Wert $|\chi(t)|$ aller Knoten t in T. Die *Baumweite* von $\mathcal{G}$ ist die kleinste Weite aller Baumzerlegungen von $\mathcal{G}$.

3 „Decomposition-guided“ (DG) Reduktionen

Für eine Vielzahl von Problemen in der KI gibt es Lösungsmethoden, die auf der Ausnutzung von Baumweite basieren, was anhand spezialisierter Implementierungen, z.B. [CW19; FHZ19; KP18], aber auch genereller frameworks [BB19; Bl16; La12] festgestellt werden kann. Gerade bei dem kanonischen Zählproblem für SAT konnte kürzlich bei der „Model Counting Competition 2021“ ein Solver gewinnen [KJ21], der Baumweite verwendet.

Motiviert durch diese Ausnutzung von Baumweite zur Problemlösung, stellt sich die Frage, wie man Instanzen zwischen unterschiedlichen Formalismen konvertieren (umwandeln) kann, sodass strukturelle Eigenschaften in der Form von Baumweite weitestgehend erhalten bleiben. Solche Reduktionen haben theoretische Relevanz aber auch praktische Anwendung. Ein Beispiel wäre die Verwendung von SAT-Solvern, indem man eine zu lösende Instanz eines Problems der KI derart in eine aussagenlogische Formel kodiert, sodass die Baumweite der Instanz linear erhalten bleibt oder zumindest nur sparsam erhöht wird. Diese Frage wird mit Hilfe von speziellen Reduktionen, die *„decomposition-guided“ (DG)*, also zerlegungsangeleitet, sind. Die Idee dieser Reduktionen ist inspiriert durch dynamische Programmierung, die Probleme mittels Traversion der Baumzerlegung von den Blättern bis zur Wurzel des Baumes in Teilen löst. Dabei werden Probleme in Teile reduziert, um präzise Baumweite-Garantien der resultierenden Instanz zu erhalten.

Das Konzept von DG-Reduktionen ist in Abbildung 1 vereinfacht dargestellt, wobei eine Instanz $\mathcal{I}$ eines Quellproblems $\mathtt{P}$ als auch eine Baumzerlegung $\mathcal{T}$ vom Graphen $\mathcal{G}_\mathcal{I}$ angenommen wird. Diese Art der Reduktion hat den Vorteil, dass sie per Konstruktion bereits neben einer Instanz des Zielproblems automatisch auch eine Baumzerlegung liefert. Des Weiteren lässt sich damit sofort eine bestimmte Weiteabhängigkeit bzw. im Idealfall sogar Baumweiteabhängigkeit des Zielproblems vom Quellproblem zeigen.

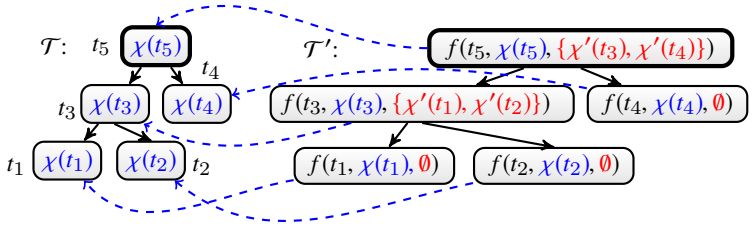


Abb. 1: Vereinfachte Darstellung einer DG-Reduktion von einem Quellproblem $\mathtt{P}$ nach einem Zielproblem $\mathtt{P}'$. Dabei nehmen wir an, dass eine Instanz $\mathcal{I}$ des Problems $\mathtt{P}$ sowie eine Baumzerlegung $\mathcal{T} = (T, \chi)$ von $\mathcal{G}_\mathcal{I}$ gegeben ist. Die Reduktion hängt sowohl von $\mathcal{I}$ als auch von der Zerlegung $\mathcal{T}$ ab und ist daher „zerlegungsangeleitet“. Sie wird für jeden Knoten t von T konstruiert und liefert auch eine Baumzerlegung $\mathcal{T}' = (T, \chi')$ von $\mathcal{G}_{\mathcal{I}'}$ der konstruierten Instanz $\mathcal{I}'$ des Problems $\mathtt{P}'$. Des Weiteren hängt $\chi'(t)$ funktional von $\chi(t)$ sowie $\chi'(t')$ aller Kindknoten t' von t ab (siehe f).

Ansatz für 2-QSAT [LM17]	**Neuer Ansatz für ℓ-QSAT**
(SAT, k)	(SAT, k)
$\downarrow$	$\downarrow$**DG-Reduktion**
(2-QSAT, $\log(n)$)	(2-QSAT, $\log(k)$)
(SAT, k)	(ℓ-QSAT, k)
$\downarrow$**?**	$\downarrow$**DG-Reduktion**
(3-QSAT, $\log(\log(n))$	((ℓ+1)-QSAT, $\log(k)$)

Tab. 2: Bestehende Beweisansätze unterer Schranken sind ähnlich zur linken Spalte, wo die EZH durch Reduktion von SAT angewendet wird. Der Parameter (Baumweite) k der Formel wird nicht direkt verwendet; der Parameter der Zielinstanz ist eine Funktion in der Variablenanzahl n der SAT Formel. Unser Ansatz ist in der rechten Spalte zu sehen, der direkt die Baumweite k der Formel via DG-Reduktion exponentiell verkleinert. Dies lässt sich daher auf Quantorenrang ℓ generalisieren.

4 Neues Meta-Werkzeug via unterer Schranken für QBFs

Für das Zeigen oberer Schranken lässt sich oft eine DG-Reduktion nach QSAT, die die Baumweite linear erhält, verwenden, um die selbe obere Schranke wie jene für QBFs, zu erhalten [Ch04]. Alternativ gibt es auch Meta-Theoreme, mit dessen Hilfe man zumindest die Existenz eines effizienten, parametrisierten Algorithmus nachweisen kann [Co90], ohne evtl. exakte obere Schranken zu erhalten. Die Literatur kennt auch bereits Einzelergebnisse für untere Schranken; darunter befinden sich auch welche für Probleme höherer Stufen der Polynomiellen Hierarchie [LM17; MM16]. Bemerkenswerterweise gibt es allerdings noch *kein Meta-Werkzeug*, das es ermöglicht, auf einfache Art und Weise untere Schranken zu zeigen. Wir adressieren diesen Mangel und stellen mit Hilfe von DG-Reduktionen und folgender präziser unterer Schranken für QSAT solch ein Werkzeug zur Verfügung.

Theorem 1 (Schranke von ℓ-QSAT) *Gegeben sei eine QBF der Form Q mit Quantorenrang ℓ, sodass die Baumweite von $\mathcal{G}_Q$ gleich k ist. Dann kann unter der Annahme der EZH die Validität von Q nicht in der Zeit* $\mathsf{tower}(\ell, o(k)) \cdot 2^{o(|\mathsf{var}(Q)|)}$ *entschieden werden.*

Der Beweisansatz von Theorem 1 ist innovativ, weil er eine DG Selbstreduktion von ℓ-QSAT auf $(\ell + 1)$-QSAT durchführt. Entgegen bestehender Beweise unterer Schranken, wird der zusätzliche Quantorenrang verwendet, um konstruktiv und präzise Baumweite exponentiell zu verkleinern. Dies ist in Tabelle 2 kurz skizziert und führt zu unteren Schranken für eine Vielzahl an Problemen der KI via DG-Reduktionen von ℓ-QSAT [He21, Tabelle 6.1].

5 Sind normale ASP Programme „härter“ als SAT?

Unter Anwendung einer DG-Reduktion von 2-QSAT und Theorem 1 kann man die folgende untere Schranke für ASP zeigen, die die bestehende obere Schranke komplettiert [JPW09].

Theorem 2 (Schranke allgemeiner Programme) *Sei ein beliebiges Programm Π gegeben, sodass die Baumweite von $\mathcal{G}_\Pi$ gleich k ist. Dann kann unter Annahme der EZH die Konsistenz von Π nicht in der Zeit* $2^{2^{o(k)}} \cdot \mathrm{poly}(|\mathsf{var}(\Pi)|)$ *entschieden werden.*

Interessanterweise ergeben sich beim Beweis von Theorem 2 keine Hürden. Das Ergebnis deckt sich auch mit den Erwartungen aus der klassischen Komplexitätstheorie, da sich das Problem auf der zweiten Stufe der Polynomiellen Hierarchie befindet [BET11].

Im Vergleich dazu sieht es bei dem Fragment der *normalen Programme* anders aus: Sowohl SAT als auch die Konsistenz normaler Programme sind NP-vollständige Probleme. Allerdings ist bekannt, dass man im Allgemeinen solche Programme nicht in eine logische Formel kodieren kann, sodass die Antwortmengen exakt über die Modelle der Formel repräsentiert werden, ohne einen subquadratischen Mehraufwand in der Anzahl der Variablen in Kauf zu nehmen [Ja06; LR06]. Nichtsdestotrotz bleibt die *Frage offen*: Ist es im Vergleich zu SAT „härter", die Konsistenz von normalen Programmen für Baumweite zu entscheiden?

Zu einem gewissen Grad lässt sich diese Frage tatsächlich affimieren, allerdings ist der zugehörige Beweis aufwändiger als Theorem 2 und die Konsequenzen weitreichender.

Theorem 3 (Schranke normaler Programme) *Sei ein beliebiges normales Programm* Π *gegeben, sodass die Baumweite von* $\mathcal{G}_\Pi$ *gleich* k *ist. Dann kann unter Annahme der EZH die Konsistenz von* Π *nicht in der Zeit* $2^{o(k \cdot \log(k))} \cdot \mathrm{poly}(|\mathsf{var}(\Pi)|)$ *entschieden werden.*

Bemerkenswerterweise betrifft hier der Mehraufwand nicht bloß die Anzahl der Variablen, sondern unter EZH ist dieses Problem für Baumweite tatsächlich aufwändiger zu lösen, als SAT. Eine informelle Begründung liegt darin, dass man auch mit kleiner Baumweite weitreichende Erreichbarkeitsprobleme oder auch transitive Abschlüsse formulieren kann, sodass die beteiligten Variablen über die gesamte Baumzerlegung verteilt sind.

Diese Beobachtungen führen zu einer Programmfamilie, die als ι*-tight* bezeichnet wird, wobei ι den Grad zwischen „tight" ($\iota = 1$) und „normal" ($\iota = k$ bei Baumweite k) angibt und daher direkter den zugrundeliegenden Lösungsaufwand widerspiegelt, vgl. Tabelle 1.

6 Ausnutzung von Baumweite trotz theoretischer Schranken

Trotz der starken unteren Schranken dieser Arbeit, ist es möglich, Solver zu entwickeln, die sogar Instanzen höherer Baumweite lösen können. Dafür analysieren wir SAT-verwandte Zählprobleme [SS10], die für quantitatives Schließen immer mehr an Bedeutung gewinnen. Die Schranken von Tabelle 1 gelten auch für diese Zählprobleme, vgl. [He21, Tabelle 6.1].

Unser Ansatz zum effizienten Ausnutzen von Baumweite liegt in der Vereinigung verschiedener Konzepte. (1) **Abstraktionen:** Wir berechnen bestimmte Abstraktionen des Primalgraphs, sodass der Lösungsprozess durch eine Baumzerlegung dieser Abstraktionen und dynamischer Programmierung angeleitet wird. (2) **Hybrides Lösen**: Handhabbare Subprobleme, die beim Lösen anhand der Abstraktion auftreten, werden an effiziente SAT-Solver übergeben, was unseren Ansatz in einen Hybriden zwischen dynamischer Programmierung und bestehenden SAT-Solvern verwandelt. (3) **Verfeinerung**: Nicht handhabbare Subprobleme sorgen für eine Verfeinerung der Abstraktionen, sodass wieder (rekursiv) dynamische Programmierung ausgeführt wird, bis gewisse Abbruchkriterien erreicht sind.

Die empirischen Ergebnisse liefern folgende Hauptbeobachtungen. Zuerst ist es bemerkenswert, dass wir Instanzen mit Baumzerlegungen lösen können, deren Weiten sogar 260 übersteigen. Der Grund hierfür liegt darin, dass der hybride Ansatz versucht, stark unstrukturierte Teile des Primalgraphs bevorzugt an bestehende SAT-Solver zu übergeben. Weiters kann man beobachten, dass für ein aufwändigeres Zählproblem, das eine doppelt exponentielle Laufzeitabhängigkeit in der Baumweite inne hat, im Vergleich nur Baumzerlegungen bis zu einer Weite von 99 gelöst werden konnten. Zwar ist dies beachtlich, zeigt jedoch auch die praktische Relevanz der Untersuchung unterschiedlicher Baumweite-Abhängigkeiten.

7 Ausblick

Diese Arbeit führt zu weiteren Forschungsfragen, die sich vor allem am steigenden Interesse von Baumweite widerspiegeln[4]. DG-Reduktionen dienen als Werkzeug für untere und obere Schranken. Allerdings sind deren Stärken, Schwächen sowie Beschränkungen oder mögliche Erweiterungen weitestgehend unerforscht. Im Bereich erklärbarer KI können sie zu beweisbaren Solver-Durchläufen für SAT-Erweiterungen, wie z.B. Zählproblemen, beitragen. Aktuell liefert Theorem 2 ein Werkzeug für untere Schranken von Baumweite, was wir aktuell für *allgemeinere Parameter* generalisieren. Jüngere Arbeiten generalisieren und erweitern dieses Ergebnis für „Constraint Programming“, was andere Ausdrücke unterer Schranken ermöglicht [FHK20], die sich aber bestimmt weiter verallgemeinern und in der Datenbanktheorie anwenden lassen. Existierende Erweiterungen der EZH erlauben eventuell noch genauere untere Schranken zu zeigen. In Anbetracht theoretischer SAT Modelle und gezeigter Zusammenhänge dieser Modelle mit Baumweite, vgl. [AFT11], erwarten wir weitere Konsequenzen der unteren Schranke von Theorem 3 in Bezug auf das interne Ausnutzen struktureller Eigenschaften von ASP-Solvern. Aktuelle Fortschritte von Zählproblemen vereinen Baumweite mit SAT-basierten Techniken, wobei Baumweite als Heuristik in den Solver kodiert wird [KJ21]. Wir sehen dabei Synergiepotenzial mit Sektion 6.

Literatur

[AFT11] Atserias, A.; Fichte, J. K.; Thurley, M.: Clause-Learning Algorithms with Many Restarts and Bounded-Width Resolution. Journal of Artificial Intelligence Research 40/, S. 353–373, 2011.

[AO14] Atserias, A.; Oliva, S.: Bounded-width QBF is PSPACE-complete. Journal of Computer and System Sciences 80/7, S. 1415–1429, 2014.

[BB19] Bannach, M.; Berndt, S.: Practical Access to Dynamic Programming on Tree Decompositions. Algorithms 12/8, S. 172, 2019.

[BET11] Brewka, G.; Eiter, T.; Truszczyński, M.: Answer set programming at a glance. Communications of the ACM 54/12, S. 92–103, 2011.

[4] „Treewidth“ liefert mehr als 21.100 Ergebnisse auf Google Scholar (Februar 2022).

[BHW21] Besin, V.; Hecher, M.; Woltran, S.: Utilizing Treewidth for Quantitative Reasoning on Epistemic Logic Programs. Theory and Practice of Logic Programming 19/5-6, S. 891–907, 2021.

[Bi09] Biere, A.; Heule, M.; van Maaren, H.; Walsh, T., Hrsg.: Handbook of Satisfiability. IOS Press, 2009.

[Bl16] Bliem, B.; Charwat, G.; Hecher, M.; Woltran, S.: D-FLAT^2: Subset Minimization in Dynamic Programming on Tree Decompositions Made Easy. Fundamenta Informaticae 147/1, S. 27–61, 2016.

[Ch04] Chen, H.: Quantified Constraint Satisfaction and Bounded Treewidth. In: ECAI'04. IOS Press, S. 161–165, 2004, ISBN: 1-58603-452-9.

[Co71] Cook, S. A.: The Complexity of Theorem-Proving Procedures. In: STOC'71. ACM, S. 151–158, 1971.

[Co90] Courcelle, B.: Graph Rewriting: An Algebraic and Logic Approach. In: Handbook of Theoretical Computer Science, Vol. B. Elsevier, S. 193–242, 1990.

[CW19] Charwat, G.; Woltran, S.: Expansion-based QBF Solving on Tree Decompositions. Fundamenta Informaticae 167/1-2, S. 59–92, 2019.

[FH21] Fandinno, J.; Hecher, M.: Treewidth-Aware Complexity in ASP: Not all Positive Cycles are Equally Hard. In: AAAI'21. S. 6312–6320, 2021.

[FHK20] Fichte, J. K.; Hecher, M.; Kieler, M. F. I.: Treewidth-Aware Quantifier Elimination and Expansion for QCSP. In: CP'20. Bd. 12333. LNCS, Springer, S. 248–266, 2020.

[FHP20] Fichte, J. K.; Hecher, M.; Pfandler, A.: Lower Bounds for QBFs of Bounded Treewidth. In: LICS'20. ACM, S. 410–424, 2020.

[FHZ19] Fichte, J. K.; Hecher, M.; Zisser, M.: An Improved GPU-Based SAT Model Counter. In: CP'19. Bd. 11802. LNCS, Springer, S. 491–509, 2019.

[Fi22] Fichte, J. K.; Hecher, M.; Thier, P.; Woltran, S.: Exploiting Database Management Systems and Treewidth for Counting. Theory and Practice of Logic Programming 22/1, S. 128–157, 2022.

[He21] Hecher, M.: Advanced Tools and Methods for Treewidth-Based Problem Solving, Doktorarbeit, Universität Potsdam, 2021, S. 184.

[He22] Hecher, M.: Treewidth-aware reductions of normal ASP to SAT - Is normal ASP harder than SAT after all? Artificial Intelligence 304/, S. 103651, 2022.

[IPZ01] Impagliazzo, R.; Paturi, R.; Zane, F.: Which Problems Have Strongly Exponential Complexity? Journal of Computer and System Sciences 63/4, S. 512–530, 2001.

[Ja06] Janhunen, T.: Some (in)translatability results for normal logic programs and propositional theories. Journal of Applied Non-Classical Logics 16/1-2, S. 35–86, 2006.

[JPW09] Jakl, M.; Pichler, R.; Woltran, S.: Answer-Set Programming with Bounded Treewidth. In: IJCAI'09. Bd. 2, S. 816–822, 2009.

[KJ21] Korhonen, T.; Järvisalo, M.: Integrating Tree Decompositions into Decision Heuristics of Propositional Model Counters (Short Paper). In: CP'21. Bd. 210. LIPIcs, Dagstuhl Publishing, 8:1–8:11, 2021.

[KL99] Kleine Büning, H.; Lettman, T.: Propositional Logic: Deduction and Algorithms. Cambridge University Press, 1999.

[KP18] Kiljan, K.; Pilipczuk, M.: Experimental Evaluation of Parameterized Algorithms for Feedback Vertex Set. In: SEA. Bd. 103. LIPIcs, Dagstuhl Publishing, 12:1–12:12, 2018.

[La12] Langer, A.; Reidl, F.; Rossmanith, P.; Sikdar, S.: Evaluation of an MSO-Solver. In: ALENEX'12. SIAM / Omnipress, S. 55–63, 2012.

[LM17] Lampis, M.; Mitsou, V.: Treewidth with a Quantifier Alternation Revisited. In: IPEC'17. Bd. 89. LIPIcs, Dagstuhl Publishing, 26:1–26:12, 2017.

[LR06] Lifschitz, V.; Razborov, A. A.: Why are there so many loop formulas? ACM Transactions on Computational Logic 7/2, S. 261–268, 2006.

[MM16] Marx, D.; Mitsou, V.: Double-Exponential and Triple-Exponential Bounds for Choosability Problems Parameterized by Treewidth. In: ICALP'16. Bd. 55. LIPIcs, Dagstuhl Publishing, 28:1–28:15, 2016.

[PV06] Pan, G.; Vardi, M. Y.: Fixed-Parameter Hierarchies inside PSPACE. In: LICS'06. IEEE Computer Society, S. 27–36, 2006.

[RS86] Robertson, N.; Seymour, P. D.: Graph Minors. II. Algorithmic Aspects of Tree-Width. Journal of Algorithms 7/3, S. 309–322, 1986.

[SS10] Samer, M.; Szeider, S.: Algorithms for propositional model counting. Journal of Discrete Algorithms 8/1, S. 50–64, 2010.

Markus Hecher wurde am 11. Mai 1990 geboren und hat seine Masterarbeit, die mit dem Würdigungspreis der Stadt Wien ausgezeichnet wurde, an der TU Wien abgeschlossen. Für einen Hauptteil seiner Doktorarbeit wurde er mit dem „Marco Cadoli Best Student Paper Award“ ausgezeichnet [He22], andere Teile sind auch publiziert, z.B. [FH21; FHP20]. Eine beteiligte Arbeit wurde mit „Among Best Papers“ zu einer Journal-Version eingeladen [Fi22]. Eine Folgearbeit wurde mit „Best Paper“ als auch ”Best Student Paper“ ausgezeichnet [BHW21]. 2021 wurde Markus an die UC Berkeley eingeladen, um ein Semester lang an einem spezialisierten SAT-Programm teilzunehmen. Für die binationale Doktorarbeit wurde Markus mit dem „Award of Excellence 2021“ des BMBWF Österreich ausgezeichnet. Er war bereits zwei Mal Mitgewinner bei wissenschaftlichen Wettbewerben. Co-Organisierte Events und detailliertere Daten sind verfügbar unter `https://dbai.tuwien.ac.at/staff/hecher/`.

Konsistenzerhaltende Evolution zusammenhängender Modelle mittels Transformationsnetzwerken[1]

Heiko Klare[2]

Abstract: Bei der Entwicklung komplexer Software-Systeme nutzen die Beteiligten verschiedene Arten von Artefakten zur Spezifikation ihrer Belange, beispielsweise Programmcode, Architekturdiagramme und Deployment-Beschreibungen. Aufgrund von Abhängigkeiten zwischen diesen Artefakten ist es essentiell ihre Konsistenz zu erhalten, um eine widerspruchsfreie Beschreibung des Systems zu erhalten. Modelltransformationen sind bereits gut untersucht und geeignet, um den Prozess der Konsistenzerhaltung für Paare von Artefakten zu automatisieren.

In der vorgestellten Arbeit untersuchen wir, wie Entwickler:innen Transformationen unabhängig und wiederverwendbar entwickeln und anschließend zu Netzwerken kombinieren können, um die Konsistenz zwischen mehr als zwei Artefakten zu erhalten. Ausgehend von einer formalen Beschreibung dieser Netzwerke leiten wir die zentralen Herausforderungen der *Synchronisation*, *Kompatibilität* und *Orchestrierung* ab. Wir entwickeln Ansätze zur Lösung dieser Herausforderungen und untersuchen deren praktische Relevanz. Weiterhin stellen wir einen Konstruktionsansatz für Netzwerke vor, mit dem Zielkonflikte bei der Optimierung von Qualitätseigenschaften reduziert werden, sowie eine Sprache, die Entwickler:innen bei der Anwendung dieses Ansatzes unterstützt.

1 Problemstellung

Für die Entwicklung eines Software-Systems nutzen Entwickler:innen und weitere Beteiligte verschiedene Sprachen, oder allgemein Werkzeuge, zur Beschreibung unterschiedlicher Belange. Meist stellt Programmcode das zentrale Artefakt dar, welches jedoch, implizit oder explizit, durch Spezifikationen von Architektur, Deployment, Anforderungen und anderen ergänzt wird. Neben der Programmiersprache verwenden die Beteiligten weitere Sprachen zur Spezifikation dieser Artefakte, beispielsweise die UML zum objektorientierten Entwurf, den OpenAPI-Standard für Schnittstellen-Definitionen, Docker für Deployment-Spezifikationen und das Palladio Component Model (PCM) für Qualitätsanalysen. Zur Erstellung einer funktionsfähigen Software müssen diese Artefakte das System einheitlich und widerspruchsfrei beschreiben. Beispielsweise müssen Dienst-Schnittstellen in allen Artefakten einheitlich repräsentiert sein. Wir sagen, die Artefakte müssen *konsistent* sein.

In der modellgetriebenen Entwicklung werden solche Artefakte allgemein als *Modelle* bezeichnet und bereits als wesentliche, zentrale Entwicklungsbestandteile genutzt, um

[1] Englischer Titel der Dissertation: „Building Transformation Networks for Consistent Evolution of Interrelated Models“ [Kl21]

[2] Karlsruher Institut für Technologie (KIT), Fakultät für Informatik, Am Fasanengarten 5, 76228 Karlsruhe, heiko.klare@kit.edu

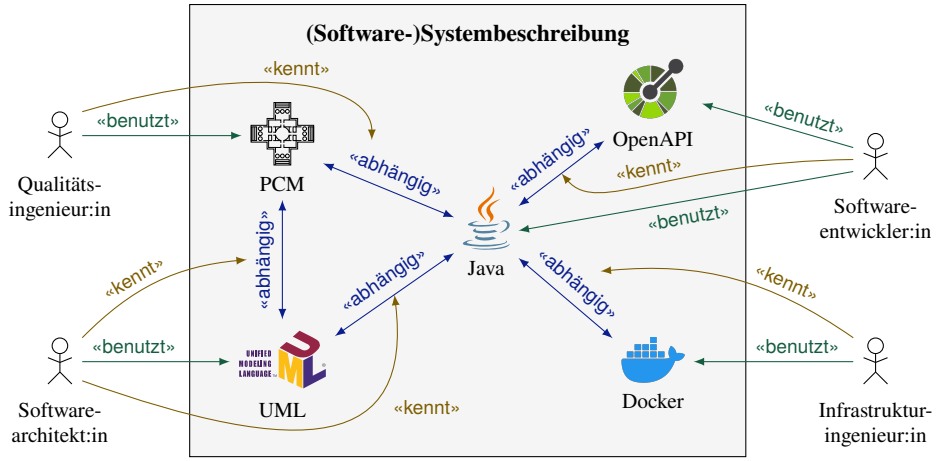


Abb. 1: Werkzeuge und Rollen in einem exemplarischen Software-Entwicklungsprozess mit verteiltem Wissen über die Beziehungen zwischen Modellen aus verschiedenen Werkzeugen (aus [Kl21, Fig. 1.1]).

auch Teile des Programmcodes aus ihnen abzuleiten. Dies betrifft beispielsweise die Softwareentwicklung für Fahrzeuge [Gu18], trifft jedoch gleichermaßen auf die allgemeine Softwareentwicklung zu, selbst wenn die Artefakte dort nicht explizit als Modelle bezeichnet werden. Ein gut untersuchtes Instrument zur Konsistenzerhaltung solcher Modelle sind *Transformationen* [Ku13], die nach Änderungen eines Modells die anderen Modelle anpassen [St10] und dadurch Konsistenz wiederherstellen. Die bisherige Forschung beschränkt sich weitestgehend auf *bidirektionale* Transformationen zur Konsistenzerhaltung zweier Modelle [St20b] und die projektspezifische Kombination von Transformationen zur Konsistenzerhaltung mehrerer Modelle [DKL18], wie ebenfalls im jüngsten Dagstuhl-Seminar zu dieser Thematik herausgestellt wurde [Cl19]. Software-Entwicklungsprojekte werden jedoch üblicherweise durch mehr als zwei Modelle beschrieben und die Sprachen, die für deren Spezifikation genutzt werden, unterscheiden sich zwischen Projekten. Jedoch werden einzelne Sprachen, wie UML-Klassendiagramme und Java-Code, über viele Projekt hinweg verwendet, sodass eine Wiederverwendung der Transformationen [Br20] zwischen diesen Sprachen wichtig ist, um den Entwicklungsaufwand zu amortisieren. Selbst ohne die Zielsetzung der Wiederverwendung ist die unabhängige Entwicklung von Transformationen vorteilhaft, denn Domänenexpertinnen und -experten, die eine Transformation spezifizieren, kennen üblicherweise nur die konsistent zu haltenden Abhängigkeiten zwischen einigen der in einem Projekt verwendeten Sprachen, aber nicht zwischen allen [Kl19]. Ein exemplarisches Entwicklungsszenario mit den darin benutzten Sprachen, Abhängigkeiten zwischen in diesen Sprachen definierten Modellen und den Kenntnissen von Domänenexpertinnen und -experten über die Abhängigkeiten stellt Abbildung 1 dar. Ein systematischer Entwicklungsprozess, in dem einzelne Transformationen unabhängig entwickelt und in unterschiedlichen Kontexten modular wiederverwendet und miteinander kombiniert werden können, existiert jedoch noch nicht und ist der Gegenstand der vorgestellten Arbeit.

2 Terminologie und Annahmen

Modelle Wir betrachten in der vorgestellten Arbeit die Konsistenzerhaltung von *Modellen*. Der zugrundeliegende Modellbegriff ist leichtgewichtig und betrachtet ein Modell als eine beliebige, nicht weiter spezifizierte Menge von Elementen. In der Softwareentwicklung betrachtete Modelle sind üblicherweise konform zur Meta Object Facility (MOF) [Ob16], unterliegen also einem objektorientierten Paradigma aus Klassen mit Attributen und Beziehungen, welches von diesem leichtgewichtigen Modellbegriff ebenfalls umfasst wird. Beispielsweise lässt sich jedes Artefakt, welches mittels XML eindeutig beschrieben werden kann, als MOF-konformes Modell auffassen. Hierunter fällt auch Programmcode.

Metamodelle Wann ein Modell valide ist, bestimmt dessen *Metamodell* [AK03]. Beispielsweise beschreibt der UML-Standard [Ob17] was ein valides UML-Modell ist oder die Java Language Specification (JLS) was ein valides Java-Programm ist. Dies kann mittels Bedingungen an die Modelle (intensional) oder durch Aufzählen der validen Modelle (extensional) definiert sein. Somit ist ein Metamodell M eine (meist unendliche) Menge von Modellen $M = \{m_1, m_2, \ldots\}$. Metamodelle sind üblicherweise in Sprachen eingebettet, mit denen diese beschrieben werden können. Sprachen umfassen zusätzlich eine konkrete Syntax zur Beschreibung (beispielsweise textuell oder grafisch) und definieren durch aufbauende Werkzeuge (z.B. einen Compiler) die Semantik der Modelle [Vö13].

Konsistenz Wir verwenden einen allgemeinen und für Modelltransformationen üblichen Konsistenzbegriff [St10]: Zwei Modelle m_1 und m_2 sind konsistent, wenn sie in einer definierten Konsistenzrelation (*consistency relation*, CR) liegen, also $\langle m_1, m_2 \rangle \in CR$. In der Praxis ist eine solche Konsistenzrelation, falls sie überhaupt explizit angegeben ist, üblicherweise intensional definiert, also mittels Bedingungen an die Modelle. Theoretisch gleich ausdrucksstark, aber praktisch üblicherweise nicht umsetzbar, ist eine extensionale Definition, bei der die konsistenten Modellpaare aufgezählt werden. Eine Konsistenzrelation zwischen UML und Java könnte beispielsweise alle UML-Klassendiagramme in Bezug zu Java-Implementierungen setzen, die die Struktur des Klassendiagramms implementieren.

Transformation Transformationen sind mathematische Funktionen, die aus Modellen andere Modell erzeugen. Inkrementelle Transformationen berücksichtigen dabei existierende Modellzustände, sodass sie Aktualisierungen nach Änderungen vornehmen können, um Konsistenz zu erhalten. Wir nutzen eine Erweiterung der üblichen Definition für *bidirektionale* Transformationen [St10]. Diese definiert zwischen zwei Metamodellen M_1 und M_2 eine Konsistenzerhaltungsregel (*consistency preservation rule*, $\textsc{Cpr}$), welche zwei Modelle annimmt und auf neue Modelle abbildet: $\textsc{Cpr} : M_1 \times M_2 \to M_1 \times M_2$. Solch eine Transformation kann beispielsweise definieren, dass nach dem Hinzufügen einer Klasse in einem UML-Diagramm eine entsprechende Java-Klasse erzeugt wird. Eine $\textsc{Cpr}$ betrachten wir als korrekt bezüglich einer zugrundeliegenden Konsistenzrelation CR, wenn die Ergebnismodelle immer konsistent sind, also das Bild $\mathrm{Im}(\textsc{Cpr}) \in CR$. Eine Konsistenzerhaltungsregel kann implizit eine Konsistenzrelation durch ihr Bild definieren, also $CR = \mathrm{Im}(\textsc{Cpr})$, und ist dann inhärent korrekt.

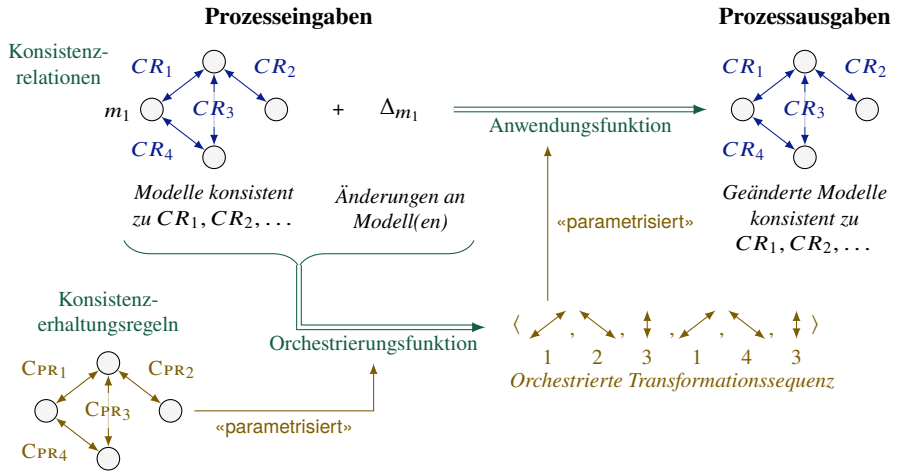


Abb. 2: Ausführungsprozess und Artefakte für modulare Konsistenzspezifikationen (aus [Kl21, Fig. 4.3]: Modelle bzw. Metamodelle (Kreise), Konsistenzrelationen (Pfeile CR_x), Konsistenzerhaltungsregeln (Pfeile CPR_x), Orchestrierungs- und Anwendungsfunktion (Doppelpfeile).

3 Anforderungen an Transformationsnetzwerke

Die Verwendung von Transformationen ist bisher auf die Konsistenzerhaltung zwischen Modellpaaren oder auf domänenspezifische Lösungen beschränkt. Wir untersuchen daher, wie Entwickler:innen mehrere Transformationen zu einem Netzwerk kombinieren können, welches die Transformationen in einer geeigneten Reihenfolge ausführen kann, sodass abschließend alle Modelle konsistent zueinander sind. Bestehende Arbeiten betrachten dieses Problem aus einer eher mathematischen Sicht, d.h. sie untersuchen unter anderem die mathematischen Anforderungen für die Dekomposition von Konsistenz zwischen mehreren Modellen in eine paarweise Betrachtung [St20b] und nehmen an, dass Transformationen aneinander angepasst werden können, um miteinander zu interoperieren [St20a].

Wir betrachten das Problem hingegen aus der Perspektive der Software-Entwicklungsmethodik und machen dabei die zentrale Annahme, dass Transformationen zwischen zwei Metamodellen unabhängig entwickelt werden und dass die Transformation nicht aneinander angepasst werden, um deren unabhängige Entwicklung und Wiederverwendung zu erlauben. Diese Annahme ergibt sich aus dem erläuterten und in Abbildung 1 skizzierten Umstand, dass Domänenexpertinnen und -experten üblicherweise nur die Abhängigkeiten für eine Teilmenge der Sprachen kennen und dass einzelne Sprachen und die Transformationen zwischen diesen über Projekt hinweg wiederverwendet werden sollen.

Basierend auf diesen Annahmen betrachten wir den in Abbildung 2 dargestellten Prozess zur Konsistenzerhaltung auf Basis von Transformationsnetzwerken. Eine Menge von Transformationen, also von Konsistenzrelationen und Konsistenzerhaltungsregeln, wird

unabhängig voneinander für Paare von Metamodellen definiert. Gegeben konkrete Modelle, die eingangs konsistent sind, sowie Änderungen an diesen Modellen, die zu potentiellen Inkonsistenzen führen, werden die Konsistenzerhaltungsregeln angewendet, um wiederum Modelle zu erhalten, die zu allen Konsistenzrelationen konsistent sind. Da die Ausführung einer Regel zwar immer lokale Konsistenz zwischen zwei Modellen herstellt, damit jedoch die Konsistenz bezüglich anderer Konsistenzrelationen verletzt werden kann, müssen die Regeln potentiell mehrfach ausgeführt werden. Hierzu betrachten wir das Konzept einer *Orchestrierungsfunktion*, die für gegebene Modelle und Änderungen eine Reihenfolge der Konsistenzerhaltungsregeln ermittelt. Eine Anwendungsfunktion wendet die so ermittelte Reihenfolge wiederum an. Wir erläutern später, dass nicht immer eine Ausführungsreihenfolge existieren muss, mit der Konsistenz wiederhergestellt werden kann, und selbst ob diese existiert ist im Allgemeinen unentscheidbar. Daher unterscheiden wir zwischen der Orchestrierungsfunktion und der Anwendungsfunktion, um die Ermittlung einer Reihenfolge davon zu entkoppeln, ob sie zu einem konsistenten Ergebnis führt.

Die vorgestellte Arbeit untersucht, wie sich ein solches Netzwerk von Transformationen korrekt erstellen lässt. Sie geht dabei von der Frage aus, wann ein solches Netzwerk als korrekt anzusehen ist und wie diese Korrektheit erreicht werden kann. Weiterhin befasst sie sich mit der Optimierung von Qualitätseigenschaften solcher Netzwerke.

4 Korrektheit von Transformationsnetzwerken

Bei der Konstruktion von Transformationsnetzwerken ist die Erwartungshaltung, dass die Transformationen für initial konsistente Modelle und Änderungen an diesen so ausgeführt werden, dass danach wieder alle Modelle konsistent sind. Wie eingangs motiviert betrachten wir einen modularen Konsistenzbegriff, der Konsistenz für eine Vielzahl von Modellen durch die Kopplung von Konsistenzrelationen und Konsistenzerhaltungsregeln zusammen mit einer Orchestrierungs- und Anwendungsfunktion beschreibt. Die Korrektheit eines Transformationsnetzwerkes entsprechend eines solchen Konsistenzbegriffs lässt sich, wie in Abbildung 3 dargestellt, in verschiedene Teilaspekte aufgliedern: (i) die Korrektheit des Netzwerkes von Konsistenzrelationen, (ii) die Korrektheit der einzelnen Konsistenzerhaltungsregeln, und (iii) die Korrektheit des Netzwerkes von Konsistenzerhaltungsregeln zusammen mit der Orchestrierungs- und Anwendungsfunktion. Aus diesen Korrektheitsbegriffen ergeben sich drei relevante Eigenschaften, die wir im Folgenden diskutieren: eine *Kompatibilitäts*-Eigenschaft für das Netzwerk an Konsistenzrelationen, eine *Synchronisations*-Eigenschaft für die einzelnen Konsistenzerhaltungsregeln, sowie das Finden einer geeigneten Ausführungsreihenfolge der Konsistenzerhaltungsregeln, einer *Orchestrierung*.

Korrektheit der Relationen Für die Menge an Konsistenzrelationen gibt es keinen natürlichen Korrektheitsbegriff, sodass sich diese, wie auch die einzelne Konsistenzrelation einer Transformation, als *korrekt per Definition* betrachten ließe. Ein Netzwerk von Konsistenzrelationen ist jedoch nicht sinnvoll, wenn es keine Modelle gibt, die zu allen Relationen konsistent sind. Dies kann beispielsweise passieren. wenn die Relationen widersprüchliche

Abb. 3: Verschiedene Korrektheitsbegriffe für Konsistenz und Konsistenzerhaltung (vereinfacht aus [Kl21, Fig. 4.4]): Metamodelle (Kreise), Konsistenz-(erhaltungs-)regeln (bidirektionale Pfeile).

Namenskonventionen für bestimmte Elemente spezifizieren, sodass Konsistenzerhaltungsregeln diese niemals gleichzeitig erfüllen könnten. In unserer Arbeit definieren wir einen darüber hinausgehenden Begriff der *Kompatibilität*, der eine wohldefinierte Art von Widerspruchsfreiheit beschreibt. Wir definieren diesen Kompatiblitätsbegriff formal, entwickeln ein formales und bewiesen korrektes Analyseverfahren, welches Kompatibilität als eine im allgemeinen unentscheidbare Eigenschaft konservativ nachweist, und leiten eine praktische Realisierung für mittels der Object Constraint Language (OCL) spezifizierte Konsistenzrelationen ab, deren Anwendbarkeit wir in Fallstudien nachweisen [Kl20].

Korrektheit der Konsistenzerhaltungsregeln Für die Korrektheit einer Konsistenzerhaltungsregel bezüglich einer Konsistenzrelation ergibt sich in Transformationsnetzwerken eine besondere Herausforderung daraus, dass beide Modelle durch die Ausführung anderer Transformationen modifiziert worden sein können und zur Wiederherstellung von Konsistenz wiederum modifiziert werden müssen (siehe auch die Terminologie in Abschnitt 2). Wurden in Anlehnung an Abbildung 1 Transformationen zwischen PCM und UML sowie zwischen PCM und Java ausgeführt, so muss die Transformation zwischen UML und Java damit umgehen, dass möglicherweise beide Modelle modifiziert wurden. Wir bezeichnen dies als *Synchronisation*. Existierende Sprachen zur Spezifikation von Transformationen beschränken sich jedoch weitestgehend auf Änderungen an einem Modell. Um die Wiederverwendung bestehender Sprachen zu erlauben, stellen wir in unserer Arbeit ein Konstruktionsverfahren für Transformationen vor, mit welchem die Synchronisations-Eigenschaft bei der Verwendung existierender Sprachen basierend auf einer formal bewiesenen Eigenschaft erfüllt wird. Dieses Verfahren basiert auf der Einsicht, dass insbesondere berücksichtigt werden muss, dass Modellelemente nicht doppelt erzeugt werden, da sie bereits durch andere Transformationen in beiden Modellen erzeugt wurden. Für dieses Verfahren zeigen wir Vollständigkeit und Angemessenheit mit einer fallstudienbasierten Evaluation in der Domäne der komponentenbasierten Softwareentwicklung. Das Verfahren erlaubt es Entwicklerinnen und Entwicklern Transformationen unabhängig und wiederverwendbar zu spezifizieren, sodass sie bei der Verwendung in einem Netzwerk korrekt miteinander interoperieren, ohne dass die anderen Transformationen im Netzwerk vorher bekannt sein müssen.

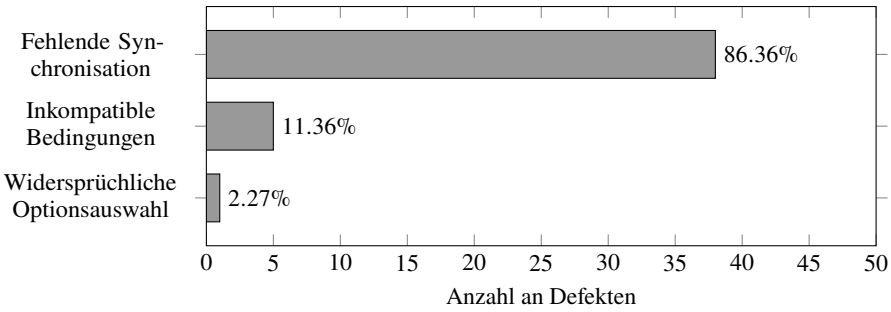


Abb. 4: Anzahl und Anteil an Defekten in Transformationsnetzwerken durch verschiedene Arten von Fehlern (vereinfacht aus [Kl21, Fig. 9.2]).

Korrektheit des Netzwerkes von Konsistenzerhaltungsregeln Die übergeordnete Korrektheitsanforderung, dass die Ausführung der Transformationen konsistente Modelle erzeugt, ist eine Eigenschaft der Konsistenzerhaltungsregeln in Kombination mit der Orchestierungs- und Anwendungsfunktion. Eine solche Ausführungsreihenfolge der Transformationen, die Konsistenz wiederherstellt, existiert nicht immer. Die Kombination von Transformationen kann dazu führen, dass eine Ausführung nie zu einem konsistenten Zustand aller Modelle konvergiert, sondern lediglich alternierende oder divergierende Modellzustände erzeugt. Die genauen Ursachen diskutieren wir in der Dissertation. Darüber hinaus kann selbst die eingeschränkte Anforderung, dass eine Konsistenz wiederherstellende Ausführungsreihenfolge der Transformationen zumindest dann gefunden werden soll, wenn sie existiert, nicht erfüllt werden. Wir zeigen, dass dieses Problem unentscheidbar ist, da sich das Halteproblem darauf reduzieren lässt [GKB21]. Weiterhin zeigen wir auf, dass die Einschränkung des Problems, um Entscheidbarkeit zu erreichen, die Anwendbarkeit unpraktikabel beschränken würde. Daher schlagen wir einen Algorithmus vor, der das Problem konservativ behandelt. Er findet eine Orchestrierung unter bestimmten, wohldefinierten Bedingungen und terminiert andernfalls mit einem Fehler. Wir beweisen die Korrektheit des Algorithmus und eine Eigenschaft, die das Finden der Ursache im Fehlerfall unterstützt.

Außerdem betrachten wir in der Arbeit welche Fehler dazu führen können, dass Korrektheit nicht gegeben ist. Wir kategorisieren dazu Fehler und untersuchen in Fallstudien wie häufig die verschiedenen Fehlerarten zu Defekten führen, wenn Transformationen kombiniert werden, die nicht für eine Verwendung in Transformationsnetzwerken entwickelt wurden. Eine Zusammenfassung stellt Abbildung 4 dar. Dabei zeigt sich, dass die meisten Defekte durch *fehlende Synchronisation* auftreten, also gerade jene Eigenschaft, für die wir ein Verfahren vorgestellt haben, mit dem diese Fehler per Konstruktion vermieden werden können. Für *inkompatible Bedingungen* haben wir ein Analyseverfahren vorgestellt, mit dem diese Fehler zumindest bei der Kopplung von Transformationen zu einem Netzwerk erkannt werden können. Fehler aus der Kategorie *widersprüchliche Optionsauswahl* können lediglich zur Laufzeit erkannt werden, waren jedoch in den Studien für den geringsten Anteil an Defekten verantwortlich und werden in der Arbeit nicht weitergehend untersucht.

(a) Vollständiger Graph (b) Baum

Abb. 5: Beispiele für Extremfälle von Netzwerktopologien (aus [Kl21, Fig. 10.1]) mit fünf Metamodellen (Knoten) und Transformationen zwischen diesen (Pfeile).

5 Qualitätseigenschaften von Transformationsnetzwerken

Neben der Korrektheit sind bei der Entwicklung von Transformationsnetzwerken auch weitere Qualitätseigenschaften, wie Wartbarkeit und Wiederverwendbarkeit, wichtig. Diese hängen zum Teil direkt von der Netzwerktopologie ab, sodass bei der Wahl einer Topologie ein Zielkonflikt bei der Optimierung von Eigenschaften wie Korrektheit und Wiederverwendbarkeit entsteht. Beispielsweise lässt sich Kompatibilität von Transformationen (als Teil der Korrektheit) erreichen, indem Zyklen im Transformationsnetzwerk vermieden werden. Dies kann durch eine Baum-Topologie erreicht werden (siehe Abbildung 5b). Die Wiederverwendbarkeit wird jedoch maximiert, wenn zwischen allen Paaren von Metamodellen Transformationen definiert sind (siehe Abbildung 5a), da nur dann beliebige Untermengen der Metamodelle wiederverwendet werden können, ohne dass dabei Transformationen zwischen diesen entfallen und damit Konsistenz nicht mehr sichergestellt werden kann. Eine Baum-Topologie reduziert jedoch die Wiederverwendbarkeit, da mit dem Weglassen von Metamodellen, die keine Blätter des Baumes sind, auch Transformationen entfernt werden.

In unserer Arbeit klassifizieren wir relevante Eigenschaften und untersuchen den Effekt verschiedener Typen von Netzwerktopologien auf diese [Kl18]. Wir leiten hieraus ein Konstruktionsverfahren für Transformationsnetzwerke ab [KG19], welches den Zielkonflikt bei der Optimierung von Qualitätseigenschaften abmildert und, unter gewissen Voraussetzungen, Korrektheit per Konstruktion gewährleistet. Dies basiert auf der Idee die Gemeinsamkeiten zwischen Metamodellen, die konsistent gehalten werden sollen, nicht in Transformationen auszudrücken, sondern in Metamodellen explizit zu machen und zu definieren, wie sich diese Gemeinsamkeiten in den konkreten Metamodellen manifestieren. Hieraus resultiert ein Netzwerk mit Baum-Topologie, in dem jedoch die konkret genutzten Metamodelle ausschließlich Blätter sind und die inneren Knoten die Metamodelle der Gemeinsamkeiten repräsentieren, wodurch die Wiederverwendbarkeit optimiert wird, während Kompatibilität aufgrund der Topologie inhärent hoch ist. Wir unterstützen den Entwicklungsprozess für diesen Ansatz mithilfe einer spezialisierten Spezifikationssprache. Während der Ansatz Zielkonflikte per Konstruktion reduziert, zeigen wir die Erfüllbarkeit der Voraussetzungen und die Vorteile der vorgeschlagenen Sprache in einer empirischen Evaluation mit einer Fallstudie aus der komponentenbasierten Softwareentwicklung.

6 Nutzen und Ausblick

Die Beiträge der vorgestellten Dissertation unterstützen sowohl Forscher:innen als auch Transformationsentwickler:innen und Softwareentwickler:innen bei der Analyse und Konstruktion von Transformationsnetzwerken. Sie stellen für Forscher:innen und Transformationsentwickler:innen systematisches Wissen über die Korrektheit und weitere Qualitätseigenschaften solcher Netzwerke bereit. Sie zeigen insbesondere, welche Teile dieser Eigenschaften per Konstruktion erreicht werden können, welche per Analyse validiert werden können, und welche Fehler unvermeidbar bei der Ausführung erwartet werden müssen. Zusätzlich zu diesen Einsichten stellen wir konkrete, praktisch nutzbare Verfahren bereit, mit denen Transformationsentwickler:innen und Softwareentwickler:innen korrekte, modular wiederverwendbare Netzwerke konstruieren, analysieren und ausführen können.

Die Arbeit basiert auf der grundlegenden Annahme, dass die Automatisierung der Konsistenzerhaltung verschiedener Entwicklungsartefakte den Entwicklungsaufwand und Fehler reduziert. Dies stellt eine sinnvolle Erwartung darstellt, sollte jedoch in Studien empirisch nachgewiesen werden. Der Vorteil durch Automatisierung muss zumindest den Aufwand für die Spezifikation der Transformationen amortisieren. Weiterhin wird in der Arbeit die nebenläufige und potentiell konfligierende Änderung von Modellen durch mehrere Benutzer:innen noch nicht betrachtet, sowie eine Transformation als vollautomatisiert angenommen. In der Praxis arbeiten jedoch verschiedene Entwickler:innen gleichzeitig an Systemen und Konsistenz kann teilweise nur durch Entscheidungen der Entwickelnden erhalten werden. Wir skizzieren in unserer Arbeit initiale Lösungsideen für diese Herausforderungen, die als Basis für die Untersuchung in zukünftigen Arbeiten dienen.

Literatur

[AK03] Atkinson, C.; Kühne, T.: „Model-driven development: a metamodeling foundation“. IEEE Software 20/5, S. 36–41, 2003.

[Br20] Bruel, J.-M.; Combemale, B.; Guerra, E.; Jézéquel, J.-M.; Kienzle, J.; de Lara, J.; Mussbacher, G.; Syriani, E.; Vangheluwe, H.: „Comparing and classifying model transformation reuse approaches across metamodels“. SoSym 19/2, S. 441–465, 2020.

[Cl19] Cleve, A.; Kindler, E.; Stevens, P.; Zaytsev, V.: „Multidirectional Transformations and Synchronisations (Dagstuhl Seminar 18491)“. Dagstuhl Reports 8/12, S. 1–48, 2019.

[DKL18] Diskin, Z.; König, H.; Lawford, M.: „Multiple Model Synchronization with Multiary Delta Lenses“. In: 21st International Conference on Fundamental Approaches to Software Engineering. Springer, S. 21–37, 2018.

[GKB21] Gleitze, J.; Klare, H.; Burger, E.: „Finding a Universal Execution Strategy for Model Transformation Networks“. In: 24th International Conference on Fundamental Approaches to Software Engineering. Springer, S. 87–107, 2021.

[Gu18] Guissouma, H.; Klare, H.; Sax, E.; Burger, E.: „An Empirical Study on the Current and Future Challenges of Automotive Software Release and Configuration Management“. In: 44th Euromicro Conference on Software Engineering and Advanced Applications. IEEE, S. 298–305, 2018.

[KG19] Klare, H.; Gleitze, J.: „Commonalities for Preserving Consistency of Multiple Models“. In: 22nd ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion. IEEE, S. 371–378, 2019.

[Kl18] Klare, H.: „Multi-Model Consistency Preservation“. In: 21st ACM/IEEE International Conference on Model Driven Engineering Languages and Systems: Companion Proceedings. ACM, S. 156–161, 2018.

[Kl19] Klare, H.; Syma, T.; Burger, E.; Reussner, R.: „A Categorization of Interoperability Issues in Networks of Transformations“. Journal of Object Technology 18/3, 12th International Conference on Model Transformations, 4:1–20, 2019.

[Kl20] Klare, H.; Pepin, A.; Burger, E.; Reussner, R.: *A Formal Approach to Prove Compatibility in Transformation Networks*, Techn. Ber. 3, Karlsruhe Reports in Informatics, Karlsruhe Institute of Technology (KIT), 2020.

[Kl21] Klare, H.: „Building Transformation Networks for Consistent Evolution of Interrelated Models“, Diss., Karlsruhe Institute of Technology (KIT), 2021.

[Ku13] Kusel, A.; Etzlstorfer, J.; Kapsammer, E.; Langer, P.; Retschitzegger, W.; Schonbock, J.; Schwinger, W.; Wimmer, M.: „A Survey on Incremental Model Transformation Approaches“. In: Workshop on Models and Evolution. CEUR-WS, S. 4–13, 2013.

[Ob16] Object Management Group (OMG): *Meta Object Facility (MOF) Core Specification*, Version 2.5.1, 2016, URL: `http://www.omg.org/spec/MOF/2.5.1/`.

[Ob17] Object Management Group (OMG): *OMG Unified Modeling Language (OMG UML)*, Version 2.5.1, 2017, URL: `https://www.omg.org/spec/UML/2.5.1/`.

[St10] Stevens, P.: „Bidirectional model transformations in QVT: semantic issues and open questions“. SoSym 9/1, S. 7–20, 2010.

[St20a] Stevens, P.: „Connecting software build with maintaining consistency between models: towards sound, optimal, and flexible building from megamodels“. SoSym 19/4, S. 935–958, 2020.

[St20b] Stevens, P.: „Maintaining consistency in networks of models: bidirectional transformations in the large“. SoSym 19/1, S. 39–65, 2020.

[Vö13] Völter, M.; Benz, S.; Dietrich, C.; Engelmann, B.; Helander, M.; Kats, L. C. L.; Visser, E.; Wachsmuth, G.: *DSL Engineering - Designing, Implementing and Using Domain-Specific Languages*. dslbook.org, 2013.

Heiko Klare, geboren 1990 in Höxter, absolvierte sein Bachelor- und Masterstudium am Karlsruher Institut für Technologie (KIT) mit Schwerpunkten in Softwaretechnik, Parallelverarbeitung und Algorithmik. Ab 2016 arbeitete er am Lehrstuhl für Software-Entwurf und -Qualität an der Extraktion impliziter Software-Modelle, an Modelltransformationssprachen und an der Kopplung von Modelltransformationen zur Konsistenzerhaltung mehrerer Artefakte. Über Letzteres promovierte er 2021 mit dem Prädikat *summa cum laude*. Seine Ergebnisse flossen in den VITRUVIUS-Ansatzes zur konsistenten, sichtenbasierte Systementwicklung und in das zugehörige Werkzeug[3] ein, dessen Entwicklung er leitet.

[3] `https://github.com/vitruv-tools`

Sicherheit und Datenschutz für Biometrische Systeme[1]

Jascha Kolberg[2]

Abstract: Biometrische Authetisierungsverfahren werden heutzutage für benutzerfreundliche Entsperrungen von mobilen Endgeräten sowie für sicherheitskritische Identifizierungsverfahren bei der Grenzkontrolle eingesetzt. Allerdings steigt auch die Anzahl der Angriffe auf biometrische Systeme mit deren Verbreitung. Daher erfordert die Bereitstellung biometrischer Systeme weiterreichende Maßnahmen um den Datenschutz gewährleisten und Missbrauch verhindern zu können. In diesem Zusammenhang werden in dieser Dissertation kryptographische Lösungen untersucht, um biometrische Daten sicher zu speichern und zudem im verschlüsselten Raum zu vergleichen. Um dabei langfristigen Schutz zu garantieren, werden ausschließlich Verfahren genutzt, die selbst zukünftigen Quantencomputern standhalten. Neben den Datenschutzbedenken wird die Sicherheit biometrischer Systeme durch Präsentationsangriffe während der Aufnahme gefährdet. Daher sind Verfahren zur Präsentationsangriff Detektierung (PAD) erforderlich um zwischen bona fiden Aufnahmen und Angriffen unterscheiden zu können. Zu diesem Zweck werden in dieser Dissertation verschiedene PAD Methoden für Fingerabdrucksysteme entwickelt und evaluiert.

1 Einleitung

Methoden zur Benutzerauthentifizierung lassen sich in drei Kategorien einteilen: Wissen, Besitz und Biometrie. Passwörter und PINs sind Beispiele für die erste Kategorie, während klassische Schlüssel, Smartcards und Token zur zweiten Kategorie gehören. Im Zusammenhang mit der letzten Kategorie wird biometrische Erkennung als die automatische Erkennung von Personen auf der Grundlage ihrer verhaltensbezogenen und biologischen Merkmale definiert [IS17]. Der Hauptvorteil der Biometrie gegenüber den beiden anderen Kategorien besteht darin, dass die biometrischen Merkmale nicht vergessen oder mit anderen geteilt werden können. Im Gegenteil, es liegt in der Natur der biometrischen Daten, dass sie nicht erneuert oder ausgetauscht werden können. In den letzten zehn Jahren wurde in den Nachrichten über mehrere Datenschutzverletzungen berichtet, bei denen biometrische Daten von mehreren Millionen Personen betroffen waren[3]. *Sicherheit und Datenschutz sind daher essenzielle Elemente biometrischer Systeme.*

[1] Englischer Titel der Dissertation: "Security Enhancement and Privacy Protection for Biometric Systems" [Ko21b]

[2] Hochschule Darmstadt, Fachbereich Informatik, Haardtring 100, 64295 Darmstadt, Deutschland
jascha.kolberg@h-da.de

[3] Nachrichtenmeldungen zu biometrischen Datenlecks (original englische Titel):
Washington Post - Hacks of OPM databases compromised 22.1 million people, federal authorities say (09.07.2015)
CNN - Hackers stole 5.6 million government fingerprints - more than estimated (23.09.2015)
Washington Post - U.S. Customs and Border Protection says photos of travelers were taken in a data breach (10.06.2019)
The Guardian - Major breach found in biometrics system used by banks, UK police and defence firms (14.08.2019)
vpnMentor - Report: Data Breach in Biometric Security Platform Affecting Millions of Users (14.08.2019)

Im Fall von Datenlecks oder gestohlenen Einträgen können die ursprünglichen biometrischen Muster rekonstruiert werden [CJ14; Ma18]. Da diese Daten Merkmale enthalten, welche eine bestimmte Person identifizieren können, ist es ebenso möglich, das ursprüngliche Muster (z. B. das aufgenommene Bild) zu rekonstruieren. Es ist wichtig zu beachten, dass diese rekonstruierten Bilder einige Verzerrungen oder Artefakte enthalten können und daher einen Menschen nicht täuschen würden. Sie sind jedoch gut genug, um das biometrische System zu täuschen, da ähnliche Merkmale extrahiert werden und der Vergleich dadurch akzeptiert wird. Ein erfolgreicher Schutzmechanismus gegen die Rekonstruktion von biometrischen Mustern erfordert eine irreversible Transformation gemäß der Definition in ISO/IEC 24745 [IS11].

Darüber hinaus ist kein Wissen erforderlich, um das biometrische Aufnahmegerät direkt anzugreifen (z. B. Präsentationsangriffe mit gedruckten Fotos oder Gummifingern). Obwohl diese Bedrohung für alle biometrischen Modalitäten besteht, konzentriert sich diese Dissertation auf Methoden zur Erkennung von Präsentationsangriffen (engl. Presentation Attack Detection, PAD) für die Fingerabdruckerkennung, da PAD-Gegenmaßnahmen stark von dem verwendeten Erfassungsgerät und damit auch von der biometrischen Modalität abhängen.

Insgesamt werden in dieser Dissertation Ansätze zur Verbesserung der Sicherheit von Fingerabdruckerkennungsanwendungen im Hinblick auf Präsentationsangriffe und weitere Methoden zum Datenschutz vorgestellt. Das übergeordnete Ziel ist es, das Vertrauen der Nutzenden sowie der Systembetreibenden für die Anwendung biometrischer Systeme zu stärken. Abgeleitet von dieser Motivation werden für diese Arbeit die folgenden Hauptforschungsfragen definiert.

Sicherheit

1. Welche Art von Daten müssen aufgenommen werden, um zuverlässig Präsentationsangriffe auf Fingerabdrucksysteme zu erkennen?

Da die mit herkömmlichen Fingerabdruckerfassungsgeräten erfassten Aufnahmen möglicherweise nicht genügend Details enthalten, um zwischen einer bona fiden Präsentation (BP) und einer Angriffspräsentation (AP) zu unterscheiden, könnten neue Erfassungsgeräte entwickelt werden.

2. Wie lassen sich automatisiert Angriffe erkennen, während die Benutzerfreundlichkeit gewährleistet bleibt?

In Anbetracht der großen Anzahl verschiedener maschineller Lernwerkzeuge ist es interessanter, mehrere Algorithmen zu vergleichen, anstatt nur einen zu verwenden.

Datenschutz

3. Welche Konzepte erfüllen die Datenschutzanforderungen für biometrische Systeme und ermöglichen zudem Echtzeitanwendungen?

Da datenschutzfreundliche Vergleiche einen erhöhten Rechenaufwand erfordern, muss sichergestellt werden, dass sichere Systeme immer noch in Echtzeit arbeiten. Die Herausforderung besteht darin, Mechanismen zu erforschen, welche die Transaktionszeit beschleunigen, ohne die biometrische Erkennungsleistung zu reduzieren. Zudem muss die kryptografische Sicherheit zwingend erhalten bleiben.

2 Erkennung von Fingerabdruck-Präsentationsangriffen

Die Entwicklung von Fingerabdruck-PAD-Methoden basiert auf einem neuen Aufnahmegerät. Dieser kamerabasierte Prototyp[4] (Abb. 1) enthält mehrere Beleuchtungseinheiten zur Aufnahme von Bildern und kurzen Videos im sichtbaren Spektrum sowie in den für Menschen unsichtbaren Nahinfrarot (NIR) und Kurzwelleninfrarot (KWIR) Spektra. Die NIR Kamera erfasst hierbei ebenfalls den sichtbaren Wellenbereich. Die Idee ist, dass Präsentations-Angriffs-Instrumente (PAIs) einen Bereich täuschen können, aber sich in den anderen Bereichen von menschlicher Haut unterscheiden. Angesichts des breiten Spektrums an PAI-Arten und PAI-Materialien [KDM18] soll das Aufnahmegerät durch Kombination aller erfassten Informationen zuverlässige Klassifizierungen ermöglichen.

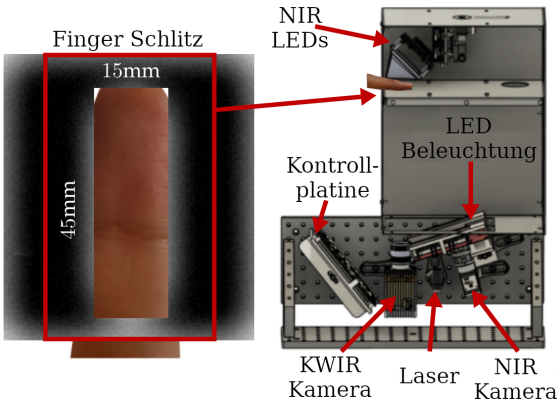


Abb. 1: Das Fingerabdruck-Aufnahmegerät wie in [Sp21] dargestellt.

Für jeden erfassten Informationskanal wurden separate PAD-Methoden entwickelt und miteinander verglichen, um den am besten geeigneten Algorithmus für jeden Datentyp zu

[4] Das Aufnahmegerät wurde während des Zeitraums der Dissertation überarbeitet. Diese Zusammenfassung bezieht sich auf die neuere Version.

finden. Diese Experimente umfassen sowohl klassische von Hand entwickelte Klassifizierungsverfahren (z. B. Support-Vektor-Maschinen) als auch neuere Deep-Learning-Ansätze (z. B. künstliche neuronale Netze). Da es von großem Interesse ist, unbekannte Angriffe zu erkennen, können Autoencoder [Ko21a] nur auf bona fiden Proben trainiert werden, um anschließend Ausreißer zu detektieren. Hierbei werden alle Darstellungen, die von den BPs in den Trainingsdaten abweichen, automatisch als APs klassifiziert. In diesem Kontext werden alle übrigen Zwei-Klassen-Klassifikatoren während des Trainierens zusätzlich auf ihre Generalisierungsfähigkeiten gegenüber unbekannten Angriffen untersucht [KGB21].

Die Datenerhebung umfasst insgesamt 24.050 Aufnahmen, die in 17.711 BPs und 4.339 APs aufgeteilt sind und von 45 verschiedenen PAI-Arten stammen. Im Allgemeinen gibt es zwei Gruppen von PAIs: ganze *Fakefinger* (z. B. Knete) und *Überschichtungen* (z. B. Silikonüberzug), die über einem echten Finger getragen werden. Darüber hinaus werden die Überschichtungen nach ihren visuellen Eigenschaften in Untergruppen eingeteilt: undurchsichtig, transparent und transluzent (teilweise durchsichtig, durchscheinend). Für die Auswertung werden mehrere Partitionen zum Trainieren und Testen der entwickelten PAD-Methoden verwendet. Zunächst umfasst die Basispartition alle PAI-Arten in den Trainings-, Validierungs- und Testdaten. Auf der Grundlage dieser Ergebnisse werden verschiedene PAD-Algorithmen fusioniert, um die fehlerhaften Klassifizierungen zu minimieren. Schließlich wird ein Protokoll zur Generalisierung definiert, so dass eine vollständige Gruppe ähnlicher PAI-Arten vom Training ausgeschlossen wird und nur in der Testmenge vorhanden ist. Insbesondere werden zunächst alle *Fakefinger* aus dem Training herausgelassen und anschließend alle *Überschichtungen*. Zusätzlich werden diese Experimente erweitert, indem auch die visuellen Untergruppen eine nach der anderen weggelassen werden. Dieses Differenzierung ermöglicht aufgrund der erwarteten Ähnlichkeit innerhalb dieser definierten (Unter-)Gruppen aussagekräftigere Ergebnisse als das Weglassen nur einer einzelnen PAI-Spezies.

Die Ergebnisse der Fusion werden in Abb. 2 anhand der Klassifikationsfehlerrate bei Angriffspräsentationen und der Klassifikationsfehlerrate bei bona fiden Präsentationen analysiert. Im Vergleich zu dem Basisszenario (durchgehend schwarz), welches keine unbekannten Angriffe enthält, ist es einfacher, unbekannte undurchsichtige (gepunktet grün) und transluzente (durchgehend gelb) Überschichtungen zu erkennen. Beide Trainingssätze enthalten jedoch nach wie vor undurchsichtige und transparente Fakefinger sowie transparente Überschichtungen. Wenn alle Fakefinger (gestrichelt blau) oder alle Überschichtungen (gestrichelt-gepunktet rot) weggelassen werden, ist die PAD-Leistung etwas schlechter als im Basisszenario. Außerdem zeigt die Grafik, dass die Erkennung unbekannter transparenter Überlagerungen (gestrichelt lila) die größte Herausforderung für diesen PAD-Ansatz darstellt.

Im Rahmen dieser Dissertation konnte gezeigt werden, dass komplementäre Informationen von mehreren Sensoren für den Klassifizierungsprozess von Vorteil sind. Mit anderen Worten: Angesichts der großen Vielfalt an PAI-Arten erzielt die Fusion bessere Ergebnisse als einzelne PAD-Algorithmen.

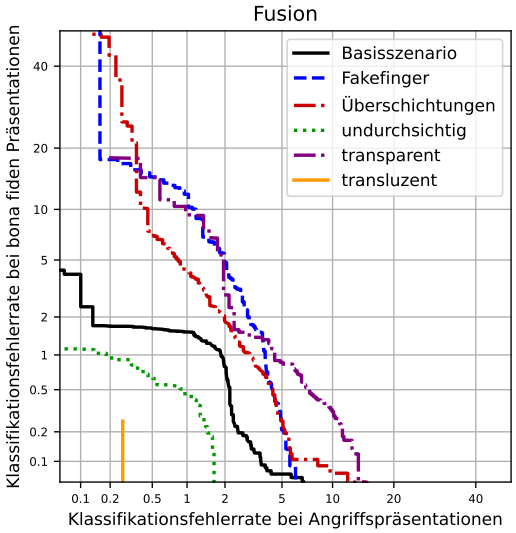


Abb. 2: Klassifikationsfehlerraten (in %) des Fusions-PAD Algorithmus auf den unterschiedlichen Partitionen. Die genannte Gruppe is jeweils nicht in den Trainingsdaten enthalten.

3 Datenschutz für biometrische Systeme

Der Standard ISO/IEC 24745 zum Schutz biometrischer Informationen [IS11] definiert drei Anforderungen für biometrische Systeme: *i)* Irreversibilität, d. h. es darf nicht möglich sein, Originalmuster aus geschützten Daten wiederherzustellen, *ii)* Unverknüpfbarkeit, d. h. es ist unmöglich, zwei geschützte Vorlagen mit derselben Person zu verknüpfen, *iii)* Erneuerbarkeit, d. h. alte Vorlagen können widerrufen und neue erstellt werden, ohne dass sich die Person erneut anmelden muss. Darüber hinaus sollte die biometrische Erkennungsleistung bei geschützten Systemen im Vergleich zu ungeschützten Systemen nicht abnehmen. Da diese Anforderungen mit zusätzlichem Rechenaufwand verbunden sind, werden in dieser Dissertation effiziente Lösungen mit langfristiger Sicherheit untersucht. Ziel ist es, generische Ansätze zu verwenden, die auf verschiedene biometrische Modalitäten anwendbar sind und in Echtzeit ausgeführt werden können. Insbesondere werden in dieser Dissertation Datenschutz Lösungen für Gesichts- und Iriserkennungssysteme entwickelt [Ko19; Ko20].

Diese Lösungen beruhen auf homomorpher Verschlüsselung (HV), die eine datenschutzgerechte Speicherung und einen Vergleich ohne Genauigkeitsverlust ermöglicht, da die Berechnungen direkt auf den Chiffretexten ausgeführt werden. Zusätzlich wird Post-Quantum-Kryptographie eingesetzt, um langfristige Sicherheit zu gewährleisten. Die Langlebigkeit biometrischer Merkmale ist der Grund, warum biometrische Daten im Vergleich zu anderen Authentifizierungsmethoden wie Passwörtern stärker geschützt werden müssen. Wenn heute geschützte biometrische Vorlagen entwendet werden, können Angreifer sie

auch in Zukunft mit Hilfe von Quantencomputern brechen, um gültige Darstellungen zu erhalten. Während Passwörter und Token nach Sicherheitsvorfällen ausgetauscht werden können, besteht bei biometrischen Systemen die Gefahr eines Imitations-Angriffs, sobald ein verwendetes Merkmal bekannt wird, da die Anzahl der biometrischen Instanzen pro Merkmal begrenzt ist (z. B. ein Gesicht, zwei Augen oder zehn Finger).

Das Konzept der HV erlaubt generell Berechnungen an Chiffretexten, ohne dass der Inhalt entschlüsselt werden muss. HV-Verfahren basieren auf der Public-Key-Kryptografie mit der Eigenschaft, dass spezielle mathematische Operationen, die auf den Chiffretext angewendet werden, direkt der entsprechenden Operation auf dem Klartext entsprechen. Für den Fall von additiven und multiplikativen Operationen sind die homomorphen Eigenschaften allgemein definiert als:

$$HV\,(A + B) = HV\,(A) \diamond HV\,(B) \tag{1}$$

$$HV\,(A \cdot B) = HV\,(A) \circ HV\,(B) \tag{2}$$

Je nach verwendetem HV-Schema können die Operationen $\diamond$ und $\circ$ variieren. Im Allgemeinen gilt, dass es eine Operation $\diamond/\circ$ gibt, die auf zwei Chiffretexte angewendet werden kann und die verschlüsselte Summe/Produkt beider Klartexte zurückgibt. Im Rahmen dieser Dissertation werden drei HV-Schemata verwendet: CKKS [Ch17], das mit Gleitkomma-Eingabedaten arbeitet, BFV [FV12], das mit ganzen Zahlen arbeitet und NTRU [HPS98] für binäre Daten.

Im Folgenden werden alle drei HV-Verfahren im Kontext eines Gesichtserkennungssystems im Hinblick auf Transaktionszeit, Dateigröße und kryptografische Sicherheit verglichen. Die erforderlichen Vorverarbeitungsschritte zur Extraktion der Gesichtsmuster sind in Abb. 3 dargestellt. Die resultierenden Gleitkomma-Vorlagen können weiter quantisiert und kodiert werden, um ganzzahlige und binäre Vorlagen zu erhalten. Somit sind die Eingabevoraussetzungen aller drei HV-Schemata erfüllt. Zur Berechnung der Distanz zwischen Probe- und Referenzvorlagen im verschlüsselten Bereich wird die quadrierte euklidische Distanz für Gleitkomma- und Integer-Darstellungen und die Hamming-Distanz für binäre Vorlagen verwendet.

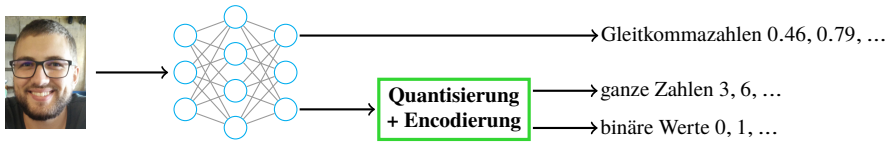


Abb. 3: Vorverarbeitungsschritte zur Extraktion der Gesichtsmuster.

Die Zeitmessungen der relevanten Transaktionen sind in Tab. 1 aufgelistet. Die Generierung der Kryptoschlüssel ist ein einmaliger Aufwand bei der Systemeinrichtung und wird bei allen Kryptosystemen innerhalb einer Sekunde abgeschlossen. Die Verschlüsselung wird einmal für jeden Referenzeintrag in der Datenbank und ein weiteres Mal für jede eingehende Probe

im Verifikations- (1:1-Vergleich) und Identifikationsmodus (1:n-Vergleich) durchgeführt. CKKS benötigt 6 ms, BFV 76 ms und NTRU 27 ms für die Verschlüsselung eines Eintrags. Der Vergleich besteht darin, den Abstand zwischen zwei geschützten Merkmalsvektoren zu berechnen, das Ergebnis zu entschlüsseln und die endgültige Entscheidung zu treffen. Ein Vergleichsverfahren mit verschlüsselten Gleitkomma-Merkmalen (3.391 ms) ist fünfmal langsamer als bei ganzzahligen Vektoren (618 ms), was wiederum von binären Darstellungen (23 ms) um das 25-fache übertroffen wird. Der einfache Vergleich zwischen zwei Einträgen bezieht sich auf die Verifikation. Um die Zeiten für die Identifikation zu erhalten, muss der gemessene Zeitwert mit der Anzahl der Referenzen in der Datenbank multipliziert werden. Da die Datenbank bereits vollständig verschlüsselt vorliegt, beeinflusst die Verschlüsselung der Probe die Gesamtzeit der Identifikation weitaus weniger als eine Verifikation.

128 bit Sicherheit	CKKS (Gleitkomma)	BFV (ganze Zahlen)	NTRU (binär)
Schlüsselgenerierung (ms)	779 (±4)	255 (±5)	362 (±84)
Verschlüsselung (ms)	6 (±2)	76 (±1)	27 (±5)
Vergleich (ms)	3,391 (±10)	618 (±26)	23 (±3)

Tab. 1: Median und Standardabweichung der Zeitmessung. Der Vergleich umfasst die Abstandsberechnung, die Entschlüsselung und basierend auf dem Schwellwert die endgültigen Entscheidung.

Zudem variieren ebenfalls die Dateigrößen, welche in Tab. 2 zusammengefasst sind. Die Unterstützung von Gleitkommazahlen erfordert mehr Speicherplatz als Ganzzahlen und Binärdateien. So werden für CKKS circa 100 MB Schlüsselmaterial erzeugt, für BFV 12 MB und für NTRU 6 KB. Im selben Schema benötigen verschlüsselte Gesichtsmerkmale in CKKS 516 KB, BFV 132 KB und NTRU 5,5 KB.

128 bit Sicherheit	CKKS (Gleitkomma)	BFV (ganze Zahlen)	NTRU (binär)
Schlüssel	99 MB	12 MB	6 KB
Merkmale	516 KB	132 KB	5.5 KB

Tab. 2: Dateigrößen der unterschiedlichen HV Systeme.

Basierend auf den Beobachtungen einzelner Instanzen, benötigt eine emulierte Datenbank mit 1.000 Einträgen etwa 500 MB im CKKS-Schema, 130 MB mit BFV und 6 MB mit NTRU. Außerdem dauert die Zeitmessung einer Identifikation für 1.000 registrierte Referenzen etwa eine Stunde mit CKKS-Verschlüsselung, fast zwölf Minuten für BFV und 23 Sekunden für NTRU.

Alle drei HV-Verfahren erreichen Irreversibilität mit Post-Quantum-Sicherheit zum langfristigen Schutz der Privatsphäre. Darüber hinaus sorgt ein Zufallsfaktor bei der Verschlüsselung dafür, dass die Chiffretexte auch bei identischen Klartexten nicht verknüpfbar sind. Generell ist die Erneuerbarkeit im spezifizierten Design möglich, indem das Schlüsselpaar ausgetauscht und die Datenbank neu verschlüsselt wird. Eine erneute Anmeldung ist nicht erforderlich, da der Client nur den öffentlichen Schlüssel verwendet.

4 Schlussbemerkungen

In dieser Dissertation wurden Methoden zur Erhöhung der Sicherheit und zum Schutz der Privatsphäre für biometrische Systeme im Zusammenhang mit Fingerabdruck-PAD und Post-Quantum sicherer Verschlüsselung untersucht. Alles in allem sind beide Bereiche, Sicherheit und Datenschutz, für biometrische Systeme von entscheidender Bedeutung. Ohne Sicherheitsmechanismen kann die Verbindung zwischen der Person und ihrer Identität gefährdet sein. Andererseits kann das Vertrauen in das System und damit die Bereitschaft, es zu nutzen, nur durch einen datenschutzgerechten Umgang mit den persönlichen sensiblen Daten erreicht werden. Basierend auf diesen Erkenntnissen können die Forschungsfragen wie folgt beantwortet werden.

1. Welche Art von Daten müssen aufgenommen werden, um zuverlässig Präsentationsangriffe auf Fingerabdrucksysteme zu erkennen?

Die Kombination von KWIR- und Laserdaten ermöglicht die Entwicklung zuverlässiger PAD-Methoden für Fingerabdrücke. Da sich die bona fiden Präsentation im Allgemeinen sehr ähnlich sind, kann eine große Anzahl von Angriffspräsentationen erfolgreich erkannt werden. Dies wiederum ermöglicht benutzerfreundliche und sichere biometrische Systeme.

2. Wie lassen sich automatisiert Angriffe erkennen, während die Benutzerfreundlichkeit gewährleistet bleibt?

Im Kontext vom diesem speziellen kamerabasierten Aufnahmegerät sind insbesondere künstliche neuronale Netze ein leistungsfähiges Werkzeug zur Verarbeitung und Klassifizierung der aufgenommenen Fotos. Der Vorteil der Verwendung von Deep-Learning-Techniken für Fingerabdruck-PAD ist die Fähigkeit, feinste Unterschiede zwischen bona fiden Präsentationen und bestimmten Angriffspräsentationen zu lernen.

3. Welche Konzepte erfüllen die Datenschutzanforderungen für biometrische Systeme und ermöglichen zudem Echtzeitanwendungen?

Alles in allem ist ein langfristiger Schutz der Privatsphäre durch sichere Post-Quantum-Kryptographie möglich. Während Verifikationen im verschlüsselten Bereich eine Echtzeit-Effizienz erreichen, sind weitere Forschungsarbeiten erforderlich, um die Identifikation im verschlüsselten Raum zu beschleunigen.

Literatur

[Ch17] Cheon, J. H.; Kim, A.; Kim, M.; Song, Y.: Homomorphic Encryption for Arithmetic of Approximate Numbers. In: Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security. Springer, S. 409–437, 2017.

[CJ14] Cao, K.; Jain, A. K.: Learning Fingerprint Reconstruction: From Minutiae to Image. IEEE Transactions on Information Forensics and Security (TIFS) 10/1, S. 104–117, 2014.

[FV12] Fan, J.; Vercauteren, F.: Somewhat Practical Fully Homomorphic Encryption. IACR Cryptology ePrint Archive 2012/, S. 144, 2012.

[HPS98] Hoffstein, J.; Pipher, J.; Silverman, J. H.: NTRU: A Ring-Based Public Key Cryptosystem. In: Proceedings of the International Algorithmic Number Theory Symposium. Springer, S. 267–288, 1998.

[IS11] ISO/IEC JTC1 SC27 Security Techniques: ISO/IEC 24745:2011. Information Technology - Security Techniques - Biometric Information Protection, 2011.

[IS17] ISO/IEC JTC1 SC37 Biometrics: ISO/IEC 2382-37:2017 Information Technology - Vocabulary - Part 37: Biometrics, 2017.

[KDM18] Kanich, O.; Drahansky, M.; Mézl, M.: Use of Creative Materials for Fingerprint Spoofs. In: Proceedings of the International Workshop on Biometrics and Forensics (IWBF). S. 1–8, 2018.

[KGB21] Kolberg, J.; Gomez-Barrero, M.; Busch, C.: On the Generalisation Capabilities of Fingerprint Presentation Attack Detection Methods in the Short Wave Infrared Domain. IET Biometrics 10/4, S. 359–373, 2021.

[Ko19] Kolberg, J.; Bauspieß, P.; Gomez-Barrero, M.; Rathgeb, C.; Dürmuth, M.; Busch, C.: Template Protection based on Homomorphic Encryption: Computationally Efficient Application to Iris-Biometric Verification and Identification. In: Proceedings of the IEEE Workshop on Information Forensics and Security (WIFS). S. 1–6, 2019.

[Ko20] Kolberg, J.; Drozdowski, P.; Gomez-Barrero, M.; Rathgeb, C.; Busch, C.: Efficiency Analysis of Post-quantum-secure Face Template Protection Schemes based on Homomorphic Encryption. In: Proceedings of the International Conference of the Biometrics Special Interest Group (BIOSIG). S. 1–4, 2020.

[Ko21a] Kolberg, J.; Grimmer, M.; Gomez-Barrero, M.; Busch, C.: Anomaly Detection with Convolutional Autoencoders for Fingerprint Presentation Attack Detection. Transactions on Biometrics, Behavior, and Identity Science (TBIOM) 3/2, S. 190–202, 2021.

[Ko21b] Kolberg, J.: Security Enhancement and Privacy Protection for Biometric Systems, Dissertation, Hochschule Darmstadt, Juli 2021.

[Ma18] Mai, G.; Cao, K.; Yuen, P. C.; Jain, A. K.: On the Reconstruction of Face Images from Deep Face Templates. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) 41/5, S. 1188–1202, 2018.

[Sp21] Spinoulas, L.; Mirzaalian, H.; Hussein, M.; AbdAlmageed, W.: Multi-Modal Fingerprint Presentation Attack Detection: Evaluation On A New Dataset. Transactions on Biometrics, Behavior, and Identity Science (TBIOM) 3/3, S. 347–364, 2021.

Jascha Kolberg wurde am 24. September 1990 in Duisburg geboren. Bereits während der Schulzeit entwickelte er ein starkes Interesse für Informatik, sodass er 2011 nach einem Praktikum bei Formtec GmbH sein Studium der IT-Sicherheit an der Ruhr-Universität Bochum anfing. Am Ende des Bachelorstudiums sammelte er weitere praktische Erfahrungen im Rahmen eines Praktikums bei der Escrypt GmbH (2014). Im anschließenden Masterstudium absolvierte Herr Kolberg ein Erasmus-Semester in Gjøvik, Norwegen, wo er u. a. den Biometriekurs bei Prof. Busch belegte. Dieser Überzeugte Herrn Kolberg, in der folgenden Masterarbeit die Themen Kryptographie und Biometrie zur Sicherung des Datenschutzes zu verknüpfen. Die Relevanz dieses Themas wurde mit dem 3. Platz beim CAST Förderpreis IT-Sicherheit gewürdigt. Motiviert durch die positive Resonanz entscheidet Herr Kolberg sich für eine Promotion an der Hochschule Darmstadt mit Fokus auf Sicherheit und Datenschutz für biometrische Systeme, welche er 2021 erfolgreich abschließt. Seitdem forscht er als Senior Wissenschaftler im Bereich Fairness für biometrische Systeme und betreut zudem bereits eine Doktorandin zum Thema Datenschutz für effiziente biometrische Identifikation. Das Foto zeigt Dr. Jascha Kolberg nach seiner Disputation am 08.07.2021.

Ein holistischer, entscheidungstheoretischer Ansatz für Pool-basiertes Aktives Lernen[1]

Daniel Kottke[2]

Abstract: Effizientes Labeling von Daten ist ein wichtiges Forschungsthema im maschinellen Lernen, da Klassifikatoren eine repräsentative Menge von gelabelten Daten benötigen um eine hohe Qualität zu erreichen. Während ungelabelte Daten leicht gesammelt werden können, ist das Labeln mühsam, zeitaufwendig oder teuer. Im sogenannten Aktiven Lernen werden Methoden entwickelt um den Aufwand des Annotationsprozesses auf ein Minimum zu reduzieren, indem nur der Teil an Daten ausgewählt wird, der den Lernfortschritt des Klassifikators vorantreibt. Diese Dissertation [Ko21a] stellt Probabilistisches Aktives Lernen vor, einen holistischen, entscheidungstheoretischen Ansatz für Pool-basiertes Lernen, das die Optimierung für jedes Gütemaß und jeden Klassifikator ermöglicht. Die ganzheitliche mathematische Beschreibung ermöglicht es, theoretische Vergleiche zu existierenden Verfahren herzustellen. Die vorgestellte Methode wird auf 22 Datensätzen für sechs verschiedene Gütemaße, sowie mehreren Klassifikatoren und die Batch-Auswahl evaluiert.

1 Einführung

Jüngste Fortschritte beim maschinellen Lernen zeigen, dass Computer in der Lage sind, komplexe Probleme zu lösen, manchmal sogar besser als Menschen. Zum Beispiel rekonstruieren Deep-Learning-Algorithmen die jeweilige Umgebung so, dass ein Auto selbstständig fahren kann [Ha20]. Auch können Methoden des maschinellen Lernens Ärzten dabei helfen, Krebs in medizinischen Bildern zu erkennen [BBS19].

Diese Methoden „erlernen“ ihre Modelle aus Beispielen. In dieser Arbeit konzentrieren wir uns auf Klassifizierungsprobleme, d. h. ein Klassifikator zielt darauf ab, Abhängigkeiten zwischen Instanzen (z. B. Bilder, numerische Merkmale von Objekten, Texte) und Klassen (z. B. gutartig vs. bösartig) zu finden. Mathematisch ausgedrückt ist ein Klassifikator $f\colon \mathbb{R}^D \to \mathcal{Y}$ eine Funktion, die eine Instanz, die durch ihren Merkmalsvektor $x \in \mathbb{R}^D$ dargestellt wird, einer Klasse $y \in \mathcal{Y} = \{1,\dots,C\}$ zuordnet, wobei D die Anzahl der Merkmale ist. Um einen guten Klassifikator zu finden, benötigen wir Trainingsdaten, die aus einer Menge von Instanz-Label-Paaren (x,y) mit $x \in \mathbb{R}^D$, $y \in \mathcal{Y}$ bestehen. Während Instanzen oft kostengünstig zur Verfügung stehen, müssen die zugehörigen Labels aufwändig beschafft werden. Zum Beispiel müssen medizinische Bilder von Fachleuten annotiert werden, die ein hohes Gehalt beziehen. Um Kosten zu sparen (z. B. Gehalt, Rechenaufwand, Gebühren oder Zeit für die Durchführung eines Experiments), zielt der Forschungsbereich des aktiven Lernens darauf ab, die Anzahl der notwendigen Labels zu reduzieren, indem nur diejenigen beschafft werden, die den Klassifikator verbessern.

[1] Englischer Titel der Dissertation: „A Holistic, Decision-Theoretic Framework for Pool-Based Active Learning“

[2] Intelligente Eingebettete Systeme, Universität Kassel, daniel.kottke@uni-kassel.de

Da effizientes Labeling zu einer zentralen Herausforderung in vielen Anwendungen geworden ist, hat sich das Gebiet des aktiven Lernens von theoretisch motivierten Konzepten hin zu mehr informationstheoretischen Ansätzen und Heuristiken entwickelt. Folglich ist das Problem oft durch den Anwendungsbereich und nicht durch eine mathematische Optimierungsfunktion gegeben. Daher funktionieren die vorgestellten Strategien in ihrem sehr eingeschränkten Kontext, aber es ist oft schwierig, diese Erkenntnisse zu nutzen, um zu erklären, wie das aktive Lernproblem gut gelöst werden kann. Oft ist es sogar unmöglich, einen experimentellen Vergleich durchzuführen, weil die getroffenen Annahmen zu einschränkend sind.

In der Dissertation werden zwei Szenarien des aktiven Lernens unterschieden:

1. Man arbeitet an der Entwicklung autonom fahrender Autos und möchte Menschen während der Fahrt aus Kamerabildern erkennen (*induktiv*).
2. Nachdem eine Naturkatastrophe einige Gebäude zerstört hat, sucht man nach Überlebenden. Daher wird anhand von Satellitenbildern klassifiziert, ob es Regionen mit eingestürzten Gebäuden gibt (*transduktiv*).

Im ersten Beispiel benötigen wir einen Klassifikator, der in der Lage ist, Menschen zu erkennen, auch wenn das Auto in Situationen unterwegs ist, in denen es bisher noch keine Daten gesammelt hat. In diesem induktiven Szenario sind die exakten Daten während der Entwicklung des Klassifikators nicht bekannt. Das zweite Beispiel beschreibt ein transduktives Szenario: Es müssen alle Bilder korrekt klassifiziert werden, da jedes Bild zerstörte Gebäude mit verletzten Personen enthalten könnte, die möglicherweise Hilfe benötigen. Obwohl wir in diesem Szenario alle Bilder im Voraus kennen, dauert das manuelle Labeln der Bilder zu lange, da Zeit ein kritischer Faktor ist. Daher müssen wir einen Klassifikator trainieren, der speziell auf die Daten zugeschnitten ist. Im Idealfall können alle „einfachen“ Fälle automatisch vom Klassifikator gelabelt werden und der Mensch kann sich auf die schwierigen Fälle konzentrieren. Auf diese Weise erhalten wir eine gute Erkennungsqualität bei geringerem Zeitaufwand, was die Rettungsaktion beschleunigt. In dieser Arbeit werden beide Szenarien behandelt und Lösungen vorgestellt.

Die zentrale Forschungsfrage der Arbeit lautet folgendermaßen: **Wie kann man einen ganzheitlichen, entscheidungstheoretischen Ansatz für aktives Lernen definieren, der eine Optimierung für jedes Gütemaß und jeden Klassifikator sowohl für induktives als auch transduktives aktives Lernen ermöglicht?**

Diese Frage wird mit einem neuen Ansatz angegangen, der als *Probabilistic Active Learning* bezeichnet wird. Zunächst werden alle möglichen Ergebnisse für jeden Kandidaten (oder jede Kandidatenmenge) simuliert und diese neuen Mengen zusammen mit den gelabelten Daten verwendet, um die erwarteten Klassenwahrscheinlichkeiten für jede beliebige Instanz mithilfe eines Bayes'schen Schätzers zu ermitteln. Ausgehend von den Vorhersagen eines Klassifikators für eine beliebige (ungelabelte) Evaluationsmenge können wir die Güte für jedes Gütemaß anhand dieser erwarteten Klassenwahrscheinlichkeiten schätzen. Anschließend wird der probabilistische Nutzen als Güteunterschied zwischen dem alten Klassifikator und dem zusätzlich auf die neuen simulierten Daten angepassten Klassifikator berechnet. Der probabilistische Nutzen dient als Wert für den Nutzen der

Auswahlstrategie. Schließlich werden die Kandidaten mit dem maximalen Nutzen für die Label-Akquisition ausgewählt.

Folgende Beiträge leitest die Dissertation [Ko21a]:

- Es wird ein ganzheitlicher, entscheidungstheoretischer Ansatz für pool-basiertes aktives Lernen vorgeschlagen, der die Auswahl für beliebige Klassifikatoren und Gütemaße sowohl im transduktiven als auch induktiven aktiven Lernszenario optimieren kann.
- Die Definition einer optimalen Selektionsstrategie, die die wahren (normalerweise unbekannten) Labels als Basisstrategie verwendet, wird in den vorliegenden Experimenten als überlegen nachgewiesen.
- Es wird gezeigt, dass das vorgeschlagene holistische Modell mehrere bestehende Strategien integriert, wenn bestimmte Annahmen hinzugefügt werden. In den Experimenten zeigt sich, dass diese zusätzlichen Annahmen die Güte der holistischen Strategie beeinträchtigen.
- Es wird eine Bewertungsmethode für transduktives aktives Lernen eingeführt, die den Kompromiss zwischen Annotations- und Fehlklassifizierungskosten visualisiert.
- Die Überlegenheit dieses Ansatzes wird für verschiedene Klassifikatoren und Gütemaße für induktives und transduktives aktives Lernen demonstriert und Experimente im Batch-basierten Szenario gezeigt.

Diese Zusammenfassung ist in drei Teile gegliedert. Zuerst wird die grundlegende Idee des Probabilistischen Aktiven Lernens präsentiert und Bezüge zur Dissertation hergestellt. Danach werden auszugsweise zentrale Ergebnisse präsentiert und diskutiert und abschließend wird ein Ausblick auf zukünftige Arbeiten gegeben.

2 Probabilistisches Aktives Lernen

Die Kernidee des Probabilistischen Aktiven Lernens besteht darin, den Güteunterschied zwischen zwei Klassifikatoren abzuschätzen: dem aktuellen Klassifikator $f^{\mathcal{L}}$, der auf die aktuell gelabelte Menge $\mathcal{L}$ trainiert wird, und einem Klassifikator $f^{\mathcal{L}^+}$, der auf einer oder mehreren zusätzlich gelabelten Instanzen angepasst wird. Vereinfachend wird in diesem Abschnitt eine konstante Batch-Größe von $b \in \mathbb{N}$ angenommen. Bezeichnen wir mit $\mathcal{L}^+ = \mathcal{L} \cup (\tilde{\mathbf{X}} \otimes \tilde{\mathbf{y}})$ die *neue* gelabelte Menge mit $\tilde{\mathbf{X}} = \langle \tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_b \rangle \in \mathcal{U}^b$ eine geordnete (Multi-)Menge von b nicht gelabelten Instanzen aus dem Kandidatenpool $\mathcal{U}$ und arbiträr zugewiesenen Labels $\tilde{\mathbf{y}} = \langle \tilde{y}_1, \ldots, \tilde{y}_b \rangle \in \mathcal{Y}^b$, die unbekannt sind, bis sie akquiriert wurden. Wir betrachten beide Mengen als geordnet, weil die erste Instanz zum ersten Label gehört, usw. Der $\otimes$ -Operator bildet paarweise Tupel aus zwei gleich großen geordneten Mengen:

$$\tilde{\mathbf{X}} \otimes \tilde{\mathbf{y}} := \langle (\tilde{\mathbf{x}}_1, \tilde{y}_1), \ldots, (\tilde{\mathbf{x}}_b, \tilde{y}_b) \rangle \quad \text{mit } \tilde{\mathbf{X}} = \langle \tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_b \rangle \text{ und } \tilde{\mathbf{y}} = \langle \tilde{y}_1, \ldots, \tilde{y}_b \rangle . \tag{1}$$

Um die Güte eines Klassifikators f zu schätzen, führen wir die empirische Güte $\mathrm{perf}_{\mathcal{E},p}(f)$ ein, die auf einer ungelabelten Evaluationsmenge $\mathcal{E}$ ermittelt wurde. Hierbei ist p eine

Abkürzung für die wahre, unbekannte Klassenzugehörigkeitswahrscheinlichkeit $p(y|x)$. Der Güteunterschied Δperf ist folgendermaßen definiert:

$$\Delta\text{perf}_{\mathcal{E},p}\left(f^{\mathcal{L}^+}, f^{\mathcal{L}}\right) = \text{perf}_{\mathcal{E},p}\left(f^{\mathcal{L}^+}\right) - \text{perf}_{\mathcal{E},p}\left(f^{\mathcal{L}}\right). \tag{2}$$

Bislang ist noch unklar, wie die Labels $\tilde{\mathbf{y}}$ ausgewählt werden, die zur Bildung von $\mathcal{L}^+$ bei der Bestimmung des Güteunterschieds erforderlich sind: Für jeden Kandidaten $\tilde{\mathbf{x}} \in \tilde{\mathbf{X}}$ können wir eines der Labels $\tilde{y} \in \mathcal{Y}$ erhalten, wenn wir das Orakel fragen. Es gibt also $|\mathcal{Y}|^b = C^b$ verschiedene Ergebnisse für die gesamte Menge $\tilde{\mathbf{X}}$. Für ein gegebenes $\tilde{\mathbf{X}}$ tritt jedes dieser Labeling-Möglichkeiten $\tilde{\mathbf{y}} = \langle \tilde{y}_1, \ldots, \tilde{y}_b \rangle \in \mathcal{Y}^b$ mit einer bestimmten Wahrscheinlichkeit auf. Um diese Wahrscheinlichkeit zu bestimmen, verwenden wir idealerweise die Verteilung $p(\tilde{y}_i|\tilde{\mathbf{x}}_i)$, die die Grundwahrheit beschreibt. Da die Grundwahrheit unbekannt ist, müssen wir sie anhand der gelabelten Menge $\mathcal{L}$ schätzen. Wir bezeichnen diese Schätzung mit $p^{\mathcal{L}}(\tilde{y}_i|\tilde{\mathbf{x}}_i) \approx p(\tilde{y}_i|\tilde{\mathbf{x}}_i)$. Für die geschätzte Verteilung, die die Wahrscheinlichkeit eines ganzen Label-Vektors $\tilde{\mathbf{y}}$ bei einer Menge von Kandidaten $\tilde{\mathbf{X}}$ beschreibt, schreiben wir $p^{\mathcal{L}}(\tilde{\mathbf{y}}|\tilde{\mathbf{X}})$ (abgekürzt durch $p^{\mathcal{L}}$). Durch Einfügen dieser Schätzung können wir den erwarteten Güteunterschied berechnen, den das Hinzufügen von $\tilde{\mathbf{X}} \otimes \tilde{\mathbf{y}}$ bringen würde:

$$\mathop{\mathbb{E}}_{p^{\mathcal{L}}(\tilde{\mathbf{y}}|\tilde{\mathbf{X}})}\left[\Delta\text{perf}_{\mathcal{E},p}\left(f^{\mathcal{L}^+}, f^{\mathcal{L}}\right)\right] = \sum_{\tilde{\mathbf{y}} \in \mathcal{Y}^b} \mathbb{p}^{\mathcal{L}}(\tilde{\mathbf{y}}|\tilde{\mathbf{X}}) \cdot \Delta\text{perf}_{\mathcal{E},p}\left(f^{\mathcal{L}^+}, f^{\mathcal{L}}\right) \tag{3}$$

Weiterhin muss noch konkretisiert werden, wie p geschätzt werden kann, um den Güteunterschied zu berechnen. Hierfür könnten wir entweder die aktuell gelabelte Menge $p^{\mathcal{L}}$ oder die neue gelabelte Menge $p^{\mathcal{L}^+}$ verwenden. Während [RM01] $p^{\mathcal{L}}$ zur Schätzung der Güte von $f^{\mathcal{L}}$ und $p^{\mathcal{L}^+}$ für $f^{\mathcal{L}^+}$ verwenden, schlagen wir vor, $p^{\mathcal{L}^+}$ zur Schätzung der Güte beider Klassifikatoren zu verwenden. Dafür gibt es folgende Argumente: (1) Bei der Verwendung verschiedener Schätzungen für die Berechnung der Güte würden wir nicht herausfinden, ob ein Unterschied auf die Änderung des Klassifikators oder auf die Änderung der Schätzungen zurückzuführen ist. (2) In dieser Arbeit gehen wir davon aus, dass das Orakel allwissend ist und daher immer richtig liegt. Folglich können wir davon ausgehen, dass mehr Labels genauere Schätzungen liefern sollten. Daher sollte $p^{\mathcal{L}^+}$ auch für den aktuellen Klassifikator $f^{\mathcal{L}}$ verwendet werden [Ko21c].

Die allgemeine Definition des probabilistischen Nutzen wird nun als Erwartungswert der Gütedifferenz über alle möglichen $\tilde{\mathbf{y}} \in \mathcal{Y}^b$ angegeben.

Definition 1 (Allgemeiner Probabilistischer Nutzen) *Bei einem aktiven Lernszenario mit ungelabelten Instanzen $\mathcal{U}$ und gelabelten Instanzen $\mathcal{L}$ definieren wir den allgemeinen probabilistischen Nutzen, indem wir den Erwartungswert der Gütedifferenz über alle möglichen Label-Möglichkeiten $\tilde{\mathbf{y}}$ für eine geordnete Kandidatenmenge $\tilde{\mathbf{X}} \subset \mathcal{U}$ wie folgt berechnen:*

$$\text{xgain}(\tilde{\mathbf{X}}, \mathcal{L}, \mathcal{E}) = \sum_{\tilde{\mathbf{y}} \in \mathcal{Y}^b} \mathbb{p}^{\mathcal{L}}(\tilde{\mathbf{y}}|\tilde{\mathbf{X}}) \cdot \Delta\text{perf}_{\mathcal{E},p^{\mathcal{L}^+}}\left(f^{\mathcal{L}^+}, f^{\mathcal{L}}\right). \tag{4}$$

Bei der Auswahlstrategie besteht die Idee darin, den probabilistischen Nutzen für verschiedene Kandidatenmengen $\tilde{\mathbf{X}} \in \mathcal{C} \subset \mathcal{U}^b$ innerhalb der Menge der Kandidatenmengen zu berechnen und diejenige für das Labeling auszuwählen, die den maximalen Nutzen erwarten lässt.

Definition 2 (Auswahl der besten Kandidaten) *Bei einem aktiven Lernszenario und einer Menge von Kandidatenmengen $\mathcal{C}$ wählen wir das/die Label für die Kandidatenmenge $\tilde{\mathbf{X}} \in \mathcal{C}$, die den probabilistischen Nutzen maximiert.*

$$\tilde{\mathbf{X}}^* = \arg\max_{\tilde{\mathbf{X}} \in \mathcal{C}} \left(\text{xgain}(\tilde{\mathbf{X}}, \mathcal{L}, \mathcal{E}) \right) \tag{5}$$

Um diesen Ansatz auf ein bestimmtes aktives Lernszenario anzuwenden, muss die obige Definition präzisiert werden, da die folgenden Fragen noch unbeantwortet sind. Diese Fragen werden in den Kapiteln 5–8 der Dissertation [Ko21a] ausführlich besprochen und diskutiert.

1. Wie kann $p^{\mathcal{L}}(y|x)$ (und $p^{\mathcal{L}^+}(y|x)$) definiert werden, um eine robuste Schätzung für $p(y|x)$ zu erhalten, ohne die Grundwahrheitsfunktion zu kennen? Robust beschreibt hier den Zustand, dass kleine Änderungen im Datensatz (z. B. durch Hinzufügen neuer gelabelter Daten) auch zu kleinen Änderungen in der Schätzung führen.
2. Wie können verschiedene Kandidatenmengen $\tilde{\mathbf{X}} \in \mathcal{C}$ erzeugt und deren Wahrscheinlichkeiten $p^{\mathcal{L}}(\tilde{\mathbf{y}}|\tilde{\mathbf{X}})$ bestimmt werden, um sequentielles, batch-basiertes und nichtmyopisches (vorausschauendes) aktives Lernen zu implementieren?
3. Wie kann Δperf für verschiedene Gütemaße berechnet werden, um eine entscheidungstheoretische Strategie zu implementieren, die in der Lage ist, das von der Anwendung vorgegebene Maß zu optimieren?
4. Wie kann man eine geeignete Evaluationsmenge $\mathcal{E}$ finden, die dem transduktiven bzw. induktiven Szenario entspricht?

3 Experimente und Ergebnisse

In der Dissertation wird gezeigt, dass mehrere bekannte Strategien so umgeschrieben werden können, dass ein theoretischer Vergleich auf der Grundlage der mathematischen Beschreibung von der hier vorgestellten Formalisierung möglich ist. Durch gezieltes Einführen von Annahmen, die von existierenden Methoden implizit getroffen wurden, kann eine Äquivalenz gezeigt werden. Details dazu befinden sich in Kapitel 11 der Dissertation [Ko21a].

Im Folgenden wird das Akquisitionsverhalten von einigen Strategien in so genannten Nutzen-Diagrammen visualisiert. Damit gehen wir über den bisherigen Forschungsansatz hinaus, aktive Lernstrategien nur durch den Vergleich ihrer Lernkurven experimentell zu evaluieren.

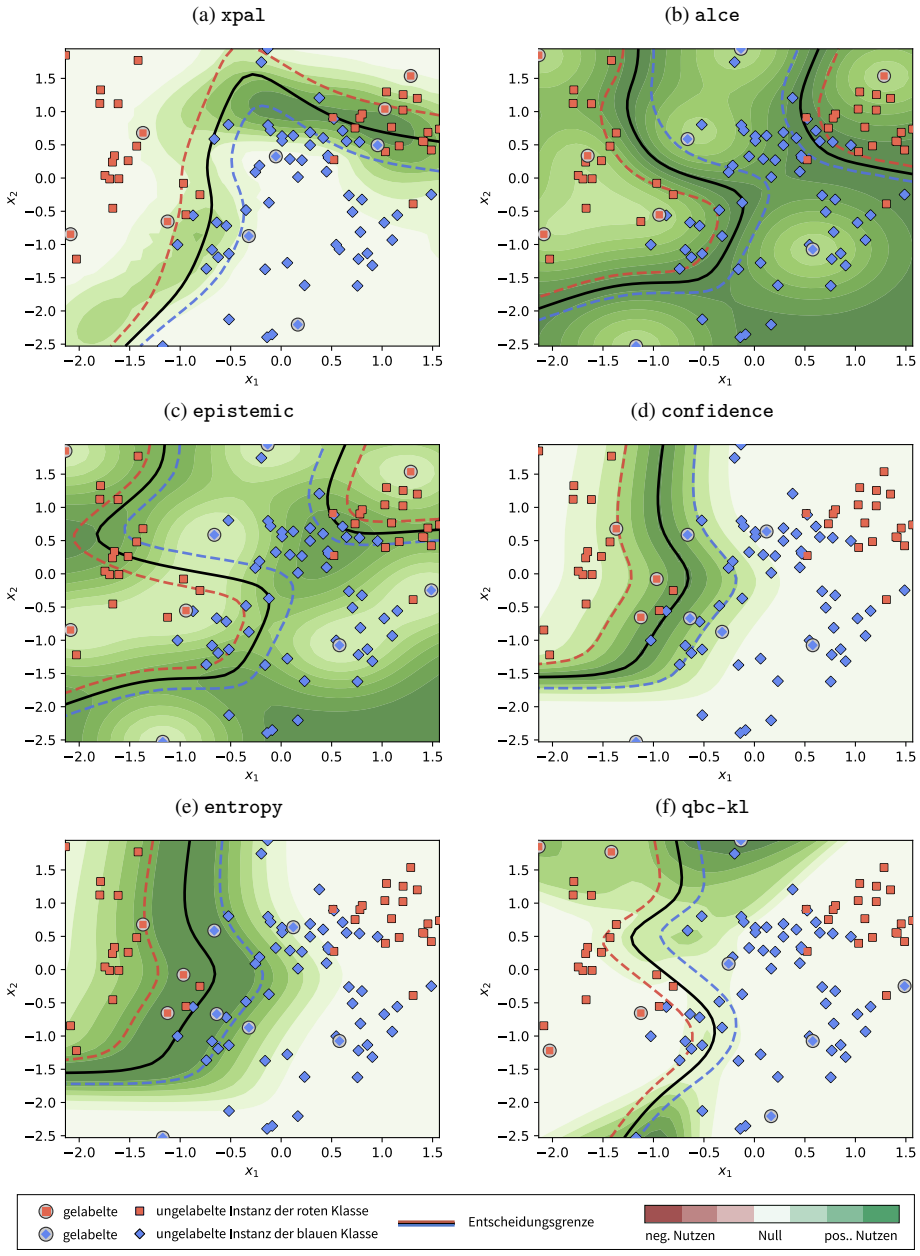


Abb. 1: Nutzen-Diagramme für die ersten 9 Labels, welche von der Selektionsstrategie selbst gewählt wurden.

Die Nutzen-Diagramme in Abb. 1 veranschaulichen, welches die ersten neun Labels sind, die von den Auswahlstrategien akquiriert werden. Weiterhin wird gezeigt, wie nützlich verschiedene Regionen für die anstehende Auswahl gesehen werden. Dafür verwenden wir einen zweidimensionalen Datensatz mit zwei Klassen (blaue Rauten vs. rote Rechtecke). Für die Klassifizierung verwenden wir einen Parzen-Window-Klassifikator [Pa62] mit einer manuell gewählten Bandbreite, so dass das Auswahlverhalten vergleichbar ist. Die Entscheidungsgrenze ist als schwarze Linie dargestellt und die gestrichelten Linien zeigen die Konfidenz des Klassifikators. Die ersten neun gelabelten Instanzen, die von der Auswahlstrategie selbst ausgewählt wurden, sind mit einem grauen Kreis markiert. Die Hintergrundfarbe zeigt, wie die jeweilige Auswahlstrategie den Nutzen eines Bereichs bewertet - dunkelgrüne Bereiche werden als nützlicher angesehen als weiße Bereiche.

Die Abbildung zeigt, dass die vorgestellte Methode xpal die ersten 9 Instanzen sehr ausgeglichen akquiriert hat und sich im Folgenden vor allem auf die Verfeinerung der Entscheidungsgrenze konzentrieren wird. Bei den unsicherheitsbasierten Methoden confidence und entropy, sowie dem Query-by-Committee-Ansatz (qbc-kl) kann man erkennen, dass die Exploration nicht ausreichend funktioniert hat, da hier die Ansammlung roter Instanzen im oberen, rechten Bereich nicht erkannt wurde. Dieses Verhalten zeigt sich auch in der folgenden quantitativen Analyse, welche anhand von Ranganalysen vorgenommen wurde.

Um die Güte der vorgenannten Auswahlstrategien quantitativ zu bewerten, führen wir Experimente an realen Datensätzen mit verschiedenen Gütemaßen, Klassifikatoren und für die Batch-Erfassung durch. Wir haben alle Algorithmen in der Python-Bibliothek `scikit-activeml`[3] [Ko21b] implementiert und veröffentlicht, die auf `scikit-learn` aufbaut. Es wurden 22 Datensätze aus der openML-Bibliothek [Va13] verwendet. Um die Experimente durchzuführen, teilen wir jeden Datensatz nach dem Zufallsprinzip in einen Trainingssatz, der aus 66% der Instanzen besteht, und einen Testsatz, der die verbleibenden 34% enthält, auf und wiederholen dies 50 Mal. Zu Beginn sind alle Instanzen ungelabelt. Da es aufgrund der großen Anzahl von Datensätzen schwierig ist, die Güte anhand von Lernkurven zu vergleichen, haben wir die Ergebnisse in Tabelle 1 zusammengefasst. Hier berechnen wir den Rang der Fläche unter der Lernkurve für jede der 50-Wiederholungen und mitteln diesen Rang für jede Kombination aus Auswahlstrategie und Datensatz. Wir verwenden Farben, um die Güte schneller erkennen zu können: Grün steht für einen guten und Rot für einen schlechten Rang. Außerdem haben wir einen Wilcoxon Signed-Rank-Test durchgeführt, um festzustellen, ob die paarweisen Unterschiede zwischen xpal und seinen Konkurrenten mit einem p-Wert von $0,01$ signifikant sind. Unten in der Tabelle wird außerdem die Anzahl der Siege, Unentschieden und Verluste auf der Basis des Tests gezählt. Tabelle 1 zeigt die Ergebnisse für das Gütemaß des Klassifikationsfehlers mit einem Parzen-Window-Klassifikator für eine sequenzielle Auswahl. Weitere Ergebnisse befinden sich in Kapitel 12 und 13 der Dissertation [Ko21a].

Die Ergebnisse zeigen, dass xpal eine leistungsstarke, gut funktionierende Auswahlstrategie ist, die auch verschiedene Gütemaße optimieren kann. Außerdem sind wir nicht auf einen bestimmten Klassifikatortyp beschränkt. Zum Beispiel konnte gezeigt werden, dass

[3] `https://github.com/scikit-activeml/scikit-activeml`

xpal auch für einen Entscheidungsbaum gut funktioniert, obwohl sich die Art der Klassifizierung von der eines kernelbasierten Verfahrens deutlich unterscheidet.

	optimal	xpal	mcpal	random	chapelle	emr	confidence	entropy	qbc_kl	alce	epistemic
iris	2.3^	3.8	2.6^	6.4v	5.9v	8.1v	7.2v	7.2v	6.4v	5.1v	-
corral	2.7^	5.1	2.9^	8.8v	10.2v	10.4v	4.5◇	4.5◇	8.4v	3.9^	4.7◇
wine	2.1^	3.3	6.3v	8.0v	3.8◇	5.6v	6.5v	5.9v	7.5v	6.0v	-
parkinsons	3.0◇	2.8	5.3v	7.7v	6.0v	7.1v	6.1v	6.1v	7.6v	6.5v	7.9v
prnn_crabs	1.0^	5.5	3.0^	8.4v	9.9v	10.2v	6.5v	6.5v	7.7v	4.0^	3.4^
sonar	2.1^	3.3	4.2v	7.6v	4.8v	7.2v	5.3v	5.3v	9.9v	8.3v	8.0v
seeds	4.2◇	3.5	2.9^	6.1v	5.1v	8.1v	7.0v	6.8v	6.1v	5.3v	-
glass	1.3^	3.0	5.5v	6.3v	7.9v	9.1v	5.0v	5.7v	6.1v	5.2v	-
qualitative-bank.	2.7^	4.9	9.9v	8.4v	9.1v	7.0v	3.9◇	3.9◇	7.0v	4.0◇	5.1◇
vertebra-column	1.7^	4.2	4.8◇	6.3v	7.2v	8.0v	5.4v	5.9v	6.0v	5.6v	-
ecoli	3.0◇	2.9	5.9v	8.3v	4.1v	6.1v	6.1v	5.7v	6.4v	6.5v	-
ionosphere	1.3^	4.5	5.2◇	9.2v	3.2^	5.0◇	7.3v	7.3v	6.8v	7.7v	8.3v
user-knowledge	1.2^	2.5	3.4v	6.6v	7.4v	9.3v	5.2v	8.5v	6.8v	4.3v	-
kc2	8.0v	5.8	5.3◇	6.2◇	5.1◇	7.0◇	5.2◇	5.0◇	5.0◇	6.4◇	7.0◇
monks-problems-1	1.3^	6.2	4.9^	8.7v	10.4v	10.6v	3.5^	3.8^	8.3v	3.8^	4.4^
balance-scale	1.0^	4.5	5.2◇	5.2◇	9.1v	8.7v	5.8v	5.1◇	4.8◇	5.7v	-
blood-transfusion	1.8^	4.3	6.2v	7.2v	8.0v	8.4v	5.9v	5.9v	5.9v	6.0v	6.5v
diabetes	1.5^	6.3	7.6v	6.1◇	6.8◇	5.8◇	5.9◇	5.9◇	4.7^	7.9v	7.5v
vehicle	1.1^	2.7	4.0v	6.0v	8.4v	8.6v	6.3v	7.8v	6.8v	3.2v	-
banknote-auth.	1.0^	4.9	2.1^	9.5v	9.2v	9.3v	5.2◇	5.2◇	9.6v	4.4◇	5.6v
car-evaluation	1.0^	3.0	2.5^	6.0v	9.0v	9.0v	7.2v	8.4v	5.4v	3.6v	-
steel-plates-fault	1.0^	2.0	4.1v	8.2v	6.2v	8.3v	4.3v	4.4v	10.6v	6.6v	10.2v
Durchschnitt	2.1	4.0	4.7	7.3	7.1	8.0	5.7	5.9	7.0	5.4	6.5
Konkurrent vs. xpal **besser**	18	0	7	0	1	0	1	1	1	3	2
gleich	3	0	4	3	3	3	5	6	2	3	3
schlechter	1	0	11	19	18	19	16	15	19	16	7

Tab. 1: Der mittlere Rang für alle Kombinationen von Auswahlstrategien und Datensätzen über 50 Wiederholungen in Bezug auf den Klassifikationsfehler unter Verwendung eines Parzen-Window-Klassifikators. Die Symbole zeigen an, ob ein Konkurrent signifikant besser ($^\wedge$), schlechter ($_\vee$) oder statistisch nicht signifikant ($\diamond$) gegenüber xpal ist, wobei ein Wilcoxon Signed-Rank-Test mit einem p-Wert von $0,01$ verwendet wurde.

4 Zusammenfassung und Ausblick

In dieser Dissertation wurde Probabilistisches Aktives Lernen vorgestellt, welches eine ganzheitliche Sichtweise der Nutzenschätzung für Auswahlstrategien bietet. Zusammenfassend lässt sich sagen, dass dieses Verfahren den Probabilistischen Nutzen berechnet, der die erwartete Güteverbesserung eines Klassifikators schätzt, wenn die Labels einiger ausgewählter, noch ungelabelter Instanzen zu seiner Trainingsmenge hinzugefügt werden.

Probabilistisches Aktives Lernen kann so spezifiziert werden, dass es für eine Vielzahl von Anwendungen und Szenarien geeignet ist. Die Experimente zeigen eine hervorragende Güte für alle genannten Varianten. Alle Entwurfsparameter (z. B. Klassifikator, Gütemaß) sind direkt durch die Anwendung vorgegeben. Neben dem methodischen Beitrag dieser Arbeit wurde das transduktive aktive Lernen formell eingeführt, ein Spezialfall, der bisher unwissentlich wie induktives Lernen behandelt wurde. Wir diskutieren, dass Selektionsstrategien das Wissen über die aktuellen Testinstanzen einbeziehen sollten, da diese in diesem Szenario bereits im Voraus bekannt sind. Darüber hinaus haben wir die minimalen aggregierten Kosten (mac) vorgeschlagen, die die Fehlklassifizierungs- und Annotationskosten zusammenfassen und einen wirtschaftlich optimalen Haltepunkt für ein gegebenes Kostenverhältnis definieren. Diese Dissertation trägt wesentlich zu einem tieferen Verständnis über aktives Lernen in dem Sinne bei, dass sie theoretische und empirische Vergleiche verschiedener Strategien liefert. Darüber hinaus zeigen wir Nutzen-Diagramme, die ein intuitives Verständnis dafür vermitteln, wie die verschiedenen Selektionsstrategien funktionieren.

Die hier vorgestellte Arbeit lässt sich in vielerlei Hinsicht erweitern, was zum Teil auch bereits in Publikationen festgehalten ist: Mehrere und unsichere Orakel [He21, Sa19], Unsicherheitsbetrachtungen für Aktives Lernen in Neuronalen Netzen [Hu21], Güteschätzungen und Abbruchkriterien [Ko19], andere Szenarien im Aktiven Lernen [Ph21, Ko16] und Anwendungen [Sc18].

Literaturverzeichnis

[BBS19] Bressan, Rafael S; Bugatti, Pedro H; Saito, Priscila TM: Breast cancer diagnosis through active learning in content-based image retrieval. Neurocomputing, 357:1–10, 2019.

[Ha20] Haussmann, Elmar; Fenzi, Michele; Chitta, Kashyap; Ivanecky, Jan; Xu, Hanson; Roy, Donna; Mittel, Akshita; Koumchatzky, Nicolas; Farabet, Clement; Alvarez, Jose M: Scalable active learning for object detection. In: Proceedings of the Intelligent Vehicles Symposium. IEEE, S. 1430–1435, 2020.

[He21] Herde, Marek; Kottke, Daniel; Huseljic, Denis; Sick, Bernhard: Multi-annotator probabilistic active learning. In: Proceedings of the International Conference on Pattern Recognition. IEEE, S. 10281–10288, 2021.

[Hu21] Huseljic, Denis; Sick, Bernhard; Herde, Marek; Kottke, Daniel: Separation of aleatoric and epistemic uncertainty in deterministic deep neural networks. In: Proceedings of the International Conference on Pattern Recognition. IEEE, S. 9172–9179, 2021.

[Ko16] Kottke, Daniel; Krempl, Georg; Stecklina, Marianne; Styp von Rekowski, Cornelius; Sabsch, Tim; Pham Minh, Tuan; Deliano, Matthias; Spiliopoulou, Myra; Sick, Bernhard: Probabilistic active learning for active class selection. In: Proceedings of the Future of Interactive Learning Machines Workshop at NIPS. 2016.

[Ko19] Kottke, Daniel; Schellinger, Jim; Huseljic, Denis; Sick, Bernhard: Limitations of assessing active learning performance at runtime. arXiv, 1901(10338), 2019.

[Ko21a] Kottke, Daniel: A Holistic, Decision-Theoretic Framework for Pool-Based Active Learning. Dissertation, University of Kassel, 2021.

[Ko21b] Kottke, Daniel; Herde, Marek; Minh, Tuan Pham; Benz, Alexander; Mergard, Pascal; Roghman, Atal; Sandrock, Christoph; Sick, Bernhard: scikit-activeml: A library and toolbox for active learning algorithms. Preprints, (2021030194):1–6, 2021.

[Ko21c] Kottke, Daniel; Herde, Marek; Sandrock, Christoph; Huseljic, Denis; Krempl, Georg; Sick, Bernhard: Toward optimal probabilistic active learning using a Bayesian approach. Machine Learning, 110:1199–1231, 2021.

[Pa62] Parzen, Emanuel: On estimation of a probability density function and mode. The Annals of Mathematical Statistics, 33(3):1065–1076, 1962.

[Ph21] Pham, Tuan; Kottke, Daniel; Krempl, Georg; Sick, Bernhard: Stream-based active learning for sliding windows under the influence of verification latency. Machine Learning, 2021. (minor revision).

[RM01] Roy, Nicholas; McCallum, Andrew: Toward optimal active learning through monte carlo estimation of error reduction. In: Proceedings of the International Conference on Machine Learning. S. 441–448, 2001.

[Sa19] Sandrock, Christoph; Herde, Marek; Calma, Adrian; Kottke, Daniel; Sick, Bernhard: Combining self-reported confidences from uncertain annotators to improve label quality. In: Proceedings of the International Joint Conference on Neural Networks. IEEE, S. 1–8, 2019.

[Sc18] Scharei, Kristina; Herde, Marek; Bieshaar, Maarten; Calma, Adrian; Kottke, Daniel; Sick, Bernhard: Automated active learning with a robot. Archives of Data Science, Series A (Online First), 5(1):1–16, 2018.

[Va13] Vanschoren, Joaquin; van Rijn, Jan N; Bischl, Bernd; Torgo, Luis: OpenML: Networked science in machine learning. SIGKDD Explorations, 15(2):49–60, 2013.

Daniel Kottke (geb. 1989) hat an der Otto-von-Guericke Universität in Magdeburg Informatik studiert und seinen Master in Data and Knowledge Engineering mit Auszeichnung abgeschlossen. Anschließend forschte er zum Thema Aktives Lernen in Magdeburg am Lehrstuhl „Knowledge Management & Discovery“ und in Kassel am Fachgebiet „Intelligente Eingebettete Systeme“. Im September 2021 schloss er seine Dissertation „A Holistic, Decision-Theoretic Framework for Pool-Based Active Learning“ mit summa cum laude ab. Aktuell ist er Gruppenleiter für „Methods for Intelligent Interactive Systems“ in Kassel. Er ist Ko-Organisator des Workshops „Interactive Adaptive Systems“ seit seiner Gründung in 2016, welcher Forscher:innen und Anwender:innen zusammenzubringen soll und ist Mitglied im Vorstand des Rats der Graduiertenakademie an der Universität Kassel.

Die Systematische Bewertung des Mehrwertes von Virtual Reality für Datenvisualisierung[1]

Matthias Kraus[2]

Abstract: Datenvisualisierung ist ein mächtiges Werkzeug, um effizient und effektiv Wissen aus Daten zu extrahieren. Für jede Kombination aus Daten und Analyseziel gibt es unterschiedliche Visualisierungstechniken, die für das jeweilige Kombination optimal sind. Doch nicht nur die Visualisierung kann die Analyse beeinflussen, sondern auch die Art und Weise, wie die Visualisierung betrachtet wird. Es macht einen großen Unterschied, ob die Daten auf den kleinen Smartphone Displays, auf Standard-Bildschirmen, oder - was eine neuere Entwicklung ist - in so genannten immersiven Umgebungen analysiert werden. In immersiven Umgebungen wird die reale Umgebung durch virtuelle Elemente erweitert oder ersetzt, was einen hohen Freiheitsgrad bei der Gestaltung von Analyseumgebungen, Visualisierungen und Interaktionskonzepten ermöglicht. Immersive Analytics umfasst Analyseverfahren, die sich solcher immersiven und ansprechenderen Medien bedienen. Eines der größten Probleme in Immersive Analytics ist derzeit, dass der Mehrwert immersiver Geräte z.B. im Vergleich zu herkömmlichen Bildschirmen nicht klar definiert ist. Es gibt kein allgmeines Regelwerk, um beispielsweise zu bestimmen, wann es sinnvoll ist, virtuelle Realität (VR) zu nutzen. Das liegt daran, dass die Technologie relativ neu ist und sich ständig weiterentwickelt. Eine regelmäßig wiederkehrende Herausforderung besteht daher darin, solche immersiven Umgebungen für Datenvisualisierung zu bewerten und die Frage zu beantworten, ob und wie der Einsatz von Virtual Reality für eine bestimmte Kombination von Daten, Aufgabe und Visualisierung sinnvoll ist.

1 Einführung & Hintergrund

Eines der größten Probleme bei Immersive-Analytics-Ansätzen besteht derzeit darin, dass der Mehrwert des Einsatzes eines immersiven Mediums, beispielsweise gegenüber einem herkömmlichen Bildschirm, nicht klar definiert ist. Eine Herausforderung besteht daher darin, immersive Umgebungen für Informationsvisualisierungsanwendungen kontinuierlich zu evaluieren und die Frage zu beantworten, ob und auf welche Weise es Sinn macht, Virtual Reality für eine bestimmte Kombination von Daten, Analyseaufgaben und Visualisierungen einzusetzen. In dieser Dissertation werden drei Strategien vorgestellt, mit deren Hilfe die Anwendbarkeit von Virtual Reality für eine bestimmte Analyse beurteilt werden kann. Die erste basiert auf logischer Argumentation mittels Ableitungen aus bestehender Literatur, um den Einsatz immersiver Medien ohne eine Evaluierung im klassischen Sinne zu rechtfertigen. Als Nebeneffekt kann dieser Ansatz Forschungslücken aufdecken und aufzeigen, welche Forschungsrichtungen vielversprechend sind und weiter untersucht werden sollten. Die zweite Strategie untersucht eine einzelne Eigenschaft, die durch immersive Umgebungen beeinflusst wird. Als beispielhafte Evaluationen werden zwei menschliche Faktoren, Immersion und Orientierung, in quantitativen Nutzerstudien betrachtet

[1] Englischer Titel der Dissertation: “Assessing the Applicability of Virtual Reality for Data Visualization”

[2] Universität Konstanz, matthias.kraus@uni-konstanz.de

und bewertet. Die dritte Strategie konzentriert sich auf die ganzheitliche Bewertung einer Immersive-Analytics-Anwendung. Die Umsetzung dieser Strategie wird anhand einer qualitativen und einer quantitativen Evaluation von zwei Anwendungen veranschaulicht. Alle drei Strategien werden im Hinblick auf die Nachhaltigkeit und Verallgemeinerbarkeit ihrer Ergebnisse diskutiert. Bislang wurde Immersive Anlytics relativ spärlich erforscht, unter anderem weil es sich hierbei um ein sich schnell entwickelndes Forschungsfeld handelt. Die in dieser Dissertation vorgestellte Forschung soll dazu beitragen, einige Forschungslücken zu schließen, indem verschiedene Ansätze aufgezeigt werden, wo und wie Virtual Reality sinnvoll eingesetzt werden kann und wo nicht.

Immersive Analytics hat in den letzten Jahrzehnten immer wieder einen Aufschwung des Forschungsinteresses erlebt. Was in den späten 90er Jahren mit immersiven CAVE-Umgebungen seinen Höhepunkt hatte, erreichte im letzten Jahrzehnt (Ende der 2010er Jahre) mit Head-Mounted Displays (HMDs) einen weiteren Höhepunkt. Dies zeigt, dass das Feld sehr vielfältig ist und nicht an eine bestimmte Technologie gebunden ist - sondern vielmehr von den spezifischen Eigenschaften der jeweiligen Technologie abhängt. Im Laufe der Jahre wurden immer mehr Anwendungsnischen gefunden, in denen der Einsatz von immersiven Umgebungen von Vorteil sein kann, z. B. Aufgaben, die stark von einer verbesserten Distanz- und Strukturwahrnehmung oder einem besseren räumliches Verständnis profitieren. Die den neuen Technologien innewohnenden Eigenschaften wie direkte Interaktion, stereoskopisches Sehen und Immersion haben sich für einige visuelle Analyseaufgaben als vorteilhaft erwiesen. Immersive Analytics ist jedoch keineswegs das Allheilmittel, das die klassischen Methoden der visuellen Analyse ablösen wird und für jedes Problem eingesetzt werden kann.

Frühere Arbeiten haben gezeigt, dass der Einsatz von immersiven Geräten mit positiven Faktoren verbunden sein kann, die wiederum einige Nachteile abmildern und die Effizienz und Effektivität bestimmter Visualisierungsansätze erhöhen könnten. Allerdings muss das Kosten-Nutzen-Verhältnis berücksichtigt werden, und nicht jeder Mangel wird automatisch behoben. So werden beispielsweise 3D-Visualisierungen, die sich auf Bildschirmen als schlecht erwiesen haben, wie 3D-Kuchendiagramme, nicht plötzlich zu guten Visualisierungswerkzeugen, nur weil sie durch Virtual-Reality (VR)-HMDs betrachtet werden. Gleichzeitig bedeutet der Einsatz immersiver Technologien nicht, dass die dargestellten Visualisierungen in 3D sein müssen oder dass die klassischen Interaktionsmodalitäten (z. B. Tastatur und Maus) aufgegeben werden müssen. Das prominenteste Vorurteil gegenüber IA ist, dass Visualisierungen in 3D dargestellt werden müssen. Immersive Analytics ist jedoch viel mehr als nur der Einsatz von 3D-Visualisierungen auf einem stereoskopischen Display. Es geht darum, Anwendungsbereiche zu finden, in denen der Einsatz immersiver Technologien im Vergleich zu einer konventionellen Visualisierung einen Mehrwert bietet oder in denen bestimmte induzierte Nachteile durch gleichzeitig induzierte Vorteile aufgewogen werden.

Daher ist es wichtig, standartisierte Wege zu finden, um die Anwendbarkeit von VR für einen bestimmten Anwendungsfall zu bewerten. Das heißt, eine Strategie, um abzuschätzen, ob der Einsatz von VR einen Mehrwert bringt oder nicht. Der Schwerpunkt dieser Arbeit liegt auf solchen Bewertungsstrategien, die sich im Vergleich zu Bewertungsansätzen auf

einer höheren Metaebene befinden. Das Ziel der Bewertung ist es, zu einer begründeten Entscheidung zu gelangen, während das Ziel der Evaluierung darin besteht, zu einer fundierten Theorie zu gelangen. Cohen et al. [CGS69] beschreiben die sog. "grounded theory" als das Ergebnis eines Bewertungsprozesses, der eine Hypothese auf der Grundlage empirischer Belege verifiziert. Im Idealfall basiert die Bewertung auf der solch einer "Grounded Theory", sie kann aber auch auf weniger als empirischen Belegen beruhen, z. B. auf logischen Überlegungen oder Ableitungen aus ähnlichen Szenarien. Allein auf dem Gebiet der Datenvisualisierung wurden viele Anstrengungen unternommen, um Bewertungsansätze zu analysieren, zu strukturieren, zu klassifizieren, zu überprüfen und zu quantifizieren. Um ein paar Beispiele zu nennen: Carpendale [Ca08] erörtert Herausforderungen bei der empirischen Validierung von Datenvisualisierung, unterteilt das Evaluationsspektrum in verschiedene Evaluationsarten und diskutiert die Vor- und Nachteile verschiedener empirischer Methodiken. Isenberg et al. [Is13] präsentieren einen systematischen Überblick über die Praxis von Evaluationen im Bereich der Visualisierung und reflektieren unter anderem Probleme, die Kreuzvalidierungen behindern und die Validität der Ergebnisse verringern. Lam et al. [La12] reflektieren sieben verschiedene Visualisierungsevaluationsansätze und diskutieren deren Vor- und Nachteile.

Das ultimative Ziel wäre es, einen allgemeingültigen Leitfaden zur Verfügung zu stellen, der empfiehlt, wann immersive Umgebungen eingesetzt werden sollten und wann nicht, nachdem alle möglichen Ansätze bewertet wurden. Im Gegensatz zur technologischen Entwicklung von Bildschirmen ist der technologische Fortschritt bei immersiven Geräten schneller und macht viel größere Sprünge, die das Erlebnis der immersiven Umgebung erheblich beeinflussen. Vor allem aufgrund des sich schnell entwickelnden Bereichs der immersiven Technologien ist es schwierig, klare Richtlinien darüber aufzustellen, wo und wie ihre Anwendung tatsächlich zu Vorteilen führt. So lassen sich beispielsweise die Ergebnisse einer Studie, in der eine bestimmte CAVE-Umgebung verwendet wurde, möglicherweise nicht auf ein aktuelles VR-HMD übertragen. Daher sollten die Ergebnisse von Evaluierungen verschiedener immersiver Umgebungen mit Vorsicht auf Situationen mit einer anderen technologischen Grundlage übertragen werden. Sie sollten bestenfalls als Ausgangspunkt für die Formulierung neuer Hypothesen über die eigene Umgebung verwendet werden. Obwohl die Forschungsanstrengungen in den letzten zehn Jahren zugenommen haben, um Antworten auf die Frage zu finden, wo und wie der Einsatz von immersiven Technologien sinnvoll ist, ist das Feld noch weitgehend unerforscht.

2 Forschungsfragen und Lösungsansatz

Ziel dieser Arbeit ist es, zu untersuchen, wie der Mehrwert von Virtual Reality für Datenvisualisierungslösungen unter Berücksichtigung der Nachteile und Einschränkungen der eingesetzten immersiven Umgebung bewertet werden kann. Die Leitfrage dieser Arbeit lautet daher:

"Wie kann der Mehrwert von virtuellen Realität für Datenvisualisierung beurteilt werden?"

Wir haben drei Möglichkeiten zur Bewertung des Mehrwertes von VR identifiziert und dementsprechend die Forschungsziele (O1-O3) dieser Arbeit formuliert. Jedes dieser Ziele befasst sich mit der Beantwortung der Frage "Wie?" auf eine bestimmte Art und Weise:

O1: durch logische Argumentation und Literaturanalyse

"Welche Argumente sprechen für oder gegen den Einsatz von VR zur Datenvisualisierung?"

Bei dieser Bewertungsstrategie werden die Meinungen durch intensive Literaturrecherche und logische Schlussfolgerungen gebildet. Das Ergebnis ist eine Reihe von unbestätigten, aber gut begründeten Hypothesen, die als Ausgangspunkt für quantitative Bewertungen verwendet werden können. Dies ist die schwächste Form der Bewertung, da bestenfalls Aussagen wie die folgenden getroffen werden können: "VR sollte bei kollaborativen Analyseaufgaben mit realistischen Avataren von Vorteil sein, da es die Kommunikation verbessert, da die Mitarbeiter sich gegenseitig sehen, im selben (virtuellen) Raum interagieren und Gestik und Mimik verwenden können". Darüber hinaus kann die Argumentation Einschränkungen enthalten, die über den derzeitigen Stand der Technik hinausgehen und daher spekulativer und hypothetischer Natur sind.

O2: durch die systematische Detailuntersuchung von Einzelaspekten

"Was unterscheidet VR von herkömmlichen Medien und welche Auswirkungen hat das?"

Der zweite Ansatz, die Anwendbarkeit von VR für ein bestimmtes Analyseziel zu bewerten, basiert auf empirischen Beweisen. Zunächst werden die Unterschiede zwischen konventionellen Medien und VR untersucht, um nach einzigartigen Merkmalen immersiver Umgebungen zu suchen, die potenziell allgegenwärtige Merkmale wie menschliche Faktoren beeinflussen. Anschließend wird ein solches Alleinstellungsmerkmal genauer untersucht, indem seine Auswirkungen auf ein bestimmtes Merkmal in einer bestimmten immersiven Analyseaufgabe bewertet werden. Bei dem Merkmal könnte es sich beispielsweise um "natürliche Körperbewegungen" in VR, das untersuchte Merkmal "Orientierung" im virtuellen Raum und die Aufgabe "Wegfindung" in der virtuellen Umgebung handeln. Typischerweise wird dies in direkten Vergleichen zwischen Bildschirm- und VR-Setups durchgeführt. Ziel sollte es sein, einzelne Faktoren zu isolieren und die Untersuchungsbedingungen am Bildschirm möglichst ähnlich zu den Bedingungen in der VR zu gestalten. Bei dieser Form der Beurteilung wird ein bestimmter Aspekt in einem bestimmten Kontext bewertet. Bei dem Versuch, Erkenntnisse aus solchen Bewertungen auf eine andere Anwendung zu übertragen, muss man sich bewusst sein, dass der veränderte Kontext einen erheblichen Einfluss auf die gewünschten Effekte haben könnte. Bei der Untersuchung von Orientierungsfähigkeiten könnte beispielsweise eine Labyrinthumgebung verwendet

werden, um die Orientierung der Benutzer beim Spielen eines Spiels auf einem Bildschirm oder in VR zu vergleichen. Das Ergebnis könnte sein, dass die Orientierung in der VR aufgrund des räumlichen Gedächtnisses besser ist. Möchte man diesen Vorteil jedoch für eine andere Aufgabe nutzen, z. B. bei einer immersiven Streudiagramm-Visualisierung, ist der Unterschied in der Orientierung zwischen Bildschirm und VR aufgrund der neuen Umgebung möglicherweise nicht groß oder sogar nicht vorhanden. Zusammenfassend lässt sich sagen, dass es schwierig ist, alle Einflussfaktoren zu bestimmen, die zu einem bestimmten Ergebnis führen, und somit die Ergebnisse zu verallgemeinern.

O3: durch die ganzheitliche Bewertung spezifischer Anwendungen

"Wie können wir die Eigenschaften von VR nutzen, um Vorteile für visuelle Analyseanwendungen zu erzielen?"

Die dritte Bewertungsstrategie ist die Bewertung einer bestimmten Anwendung auf höchster Ebene. Während es möglich sein kann, einzelne Aspekte der Anwendung zu vergleichen, wenn sie über verschiedene Medien wahrgenommen werden, ist es oft nicht möglich, die gesamte Anwendung in einer direkten Gegenüberstellung zu vergleichen. Selbst bei Techniken und Anwendungen, bei denen ein Vergleich mit einer analogen, bildschirmbasierten Anwendung oder Technik möglich ist, ist ein quantitativer Vergleich fraglich, da viele Faktoren verändert werden und das "analoge" System völlig anders aufgebaut sein könnte. Daher konzentriert sich diese Bewertungsstrategie hauptsächlich auf eine qualitative Bewertung der immersiven Umgebung für eine bestimmte Aufgabe und kann aufzeigen, wo der Einsatz von VR zu vielversprechenden Ergebnissen führt. Im Gegensatz zu Einzelaspektevaluationen hat dieser Ansatz den Vorteil, dass die Anwendbarkeit von VR für ein bestimmtes Analyseziel direkt bewertet werden kann, aber den Nachteil, dass es nicht sehr kontrollierbar ist, ob aufgedeckte Vorteile direkt auf immersive Umgebungen zurückgeführt werden können.

Zusammenfassend kann man sagen, dass es drei verschiedene Strategien mit unterschiedlichen Stärken und Schwächen gibt, um die Anwendbarkeit von VR-Umgebungen zu bewerten. In meiner Dissertation wird jede Strategie anhand mehrerer Beispiele demonstriert.

3 Beiträge, Erkentnisse und Perspektiven

Die leitende Forschungsfrage dieser Arbeit lautet "Wie kann der Mehrwert von Virtual Reality für Datenvisualisierung bewertet werden?". Der gewählte Term der "Bewertung" ist nicht strikt auf Evaluationsstrategien und -ansätze beschränkt, sondern zielt allgemein darauf ab, wie man beurteilen kann, ob und wie der Einsatz von VR für eine bestimmte Anwendung sinnvoll ist. Zur Beantwortung der Forschungsfrage wurde das Bewertungsspektrum in drei Arten von Strategien unterteilt, die in drei Kapiteln diskutiert und jeweils

mit mindestens zwei Beispielen illustriert werden. Während die Diskussion der Gesamtheit der Assessments und deren Kategorisierung zu Erkenntnissen auf einer Metaebene führt, liegt der größte Beitrag dieser Arbeit in den einzelnen Assessments, die vielfältige Erkenntnisse in unterschiedlichen Forschungsrichtungen präsentieren.

Bildung einer Argumentativen Grundlage.
Nach einem Überblick über verwandte Arbeiten bildet die Arbeit zunächst eine allgemeine argumentative Grundlage für die Forschung im Bereich "Immersive Analytics". Die erste Art von Bewertungsstrategie, die dort vorgestellt wird, baut auf eine Sammlung von Argumenten aus der Literatur und logische Ableitungen, um für und gegen den Einsatz von immersiven Umgebungen zu Analysezwecken zu argumentieren und zu zeigen, wie man sie einsetzt. Insgesamt werden hier drei Bewertungsbeispiele angeführt (s. Abbildung 1). Das erste Beispiel argumentiert, dass VR für bestimmte Schritte der Analysevalidierung von Vorteil sein könnte. Im zweiten Beispiel wird die Diskussion dann auf einer höheren Ebene fortgesetzt bei dem der zentrale Gedanke ist: Was würde passieren, wenn wir das herkömmliche Medium "Bildschirm" in einer typischen, generischen visuellen Analysepipeline durch ein VR-Medium ersetzen würden? Dabei gehen wir zunächst von einer optimalen VR-Umgebung aus, um uns dann schrittweise an ein realistisches Szenario anzunähern. Aus der Diskussion schließen wir, dass - zumindest in der Theorie - VR den herkömmlichen Medien in nichts nachsteht und sogar einzigartige Möglichkeiten bietet. Aufgrund technologischer Unzulänglichkeiten und typischerweise abweichender Nutzungen immersiver Umgebungen ist dies jedoch häufig nicht der Fall, und es muss in jedem Einzelfall geprüft werden, ob der Einsatz von VR sinnvoll ist.

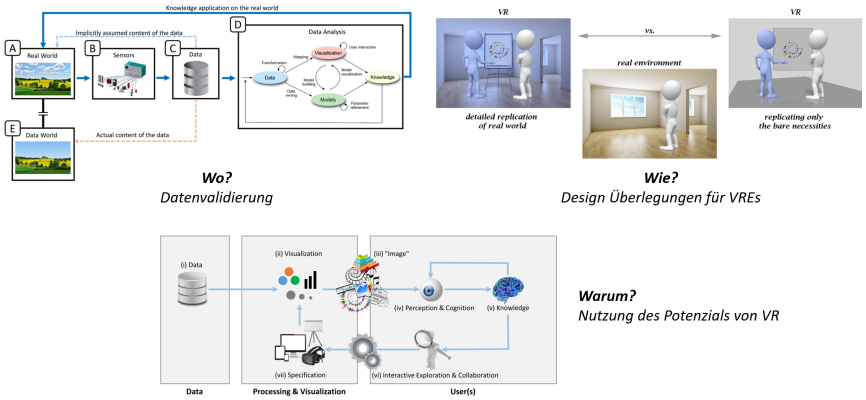


Abb. 1: Bildung einer argumentativen Grundlage basierend auf logischen Ableitungen und Literatur. Die Dissertation führt drei Beispiele an wie solch eine Bewertung aussehen kann.

Im letzten Bewertungsbeispiel wird der Schwerpunkt von der Frage, ob es sinnvoll ist, VR zur Datenvisualisierung einzusetzen, auf die Frage verlagert, wie die VR-Umgebung gestaltet werden sollte. In diesem Abschnitt wird insbesondere diskutiert, ob es sinnvoll ist, die reale Welt in der virtuellen Umgebung so weit wie möglich zu replizieren oder nicht. Dabei kommen wir zu dem Schluss, dass es nicht verallgemeinerbar ist, wie viel Nachbildung der realen Welt wünschenswert ist, und dass es stark vom Szenario abhängt, ob und inwieweit eine solche Nachbildung für eine bestimmte Anwendung vorteilhaft sein

kann. Unsere Ergebnisse sind zwar zuverlässig, allgemein anwendbar und motivieren zu weiteren Forschungen in diesem Bereich, sie weisen aber auch verschiedene nicht belegte Hypothesen auf, die zusätzliche Forschung erfordern, um ihre Kernaussagen zu festigen. Künftige Arbeiten könnten den vorgestellten Argumentationsrahmen erweitern, weitere Leitlinien erörtern, aber auch aufgestellten Hypothesen quantitative verifizieren.

Systematische Detailuntersuchung von Einzelaspekten.
Im nächsten Kapitel liegt der Fokus auf der zweiten Bewertungsstrategie. Bei diesem Ansatz werden kontrollierte Studien zu Eigenschaften, Merkmalen und Bedingungen immersiver Umgebungen durchgeführt, um Einblicke in das rudimentäre Zusammenspiel von Ursache und Wirkung zu gewinnen. Ziel könnte es beispielsweise sein, den Einfluss der Immersion, einer einzigartigen Eigenschaft immersiver Umgebungen, auf das Erinnerungsvermögen zu bewerten. Wird ein starker Effekt festgestellt, könnten diese Eigenschaften in zukünftigen Anwendungen systematisch genutzt werden, um ihre Auswirkungen auf die gesteigerte Gedächtnisleistung der Benutzer auszunutzen. Die Arbeit führt zwei Beispiele für solche Bewertungen an (s. Abbildung 2).

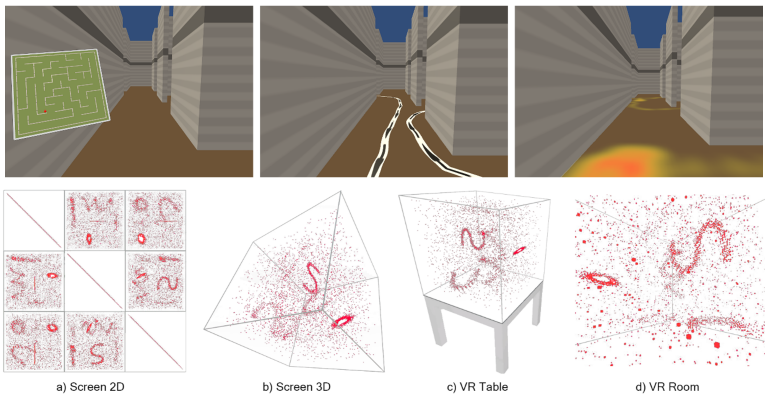


Abb. 2: Zwei Beispiele zur Einzelaspektanalyse in immersiven Umgebungen. Eine Studie untersucht die Auswirkung von visuellen Orientierungshilfen auf Benutzer, während eine Zweite den Einfluss von Immersion auf Cluster-Identifizierungs-Aufgaben bewertet.

Die erste vergleicht im Wesentlichen einen herkömmlichen Bildschirm mit einer VR Umgebung für die Analyse von Streudiagrammen, wobei vier Bedingungen mit zunehmendem Immersionsgrad verwendet wurden. In der quantitativen Benutzerstudie bearbeiteten die Teilnehmer Aufgaben zur Identifizierung von Clustern in 2D-Streudiagramm-Matrizen und 3D-Streudiagrammen. Unsere Ergebnisse zeigen, dass sich mit zunehmendem Immersionsgrad die Orientierung und das Erinnerungsvermögen verbesserten und die Lernkurven abflachten, während der Interaktionsaufwand und die Dauer der Aufgabendurchführung zunahmen. Bei der Gestaltung zukünftiger Analyseumgebungen für ähnliche Visualisierungen und Aufgaben können unsere Ergebnisse berücksichtigt werden. Wenn beispielsweise eine Aufgabe zu einer 3D-Streudiagramm-Visualisierung von einem Benutzer gute Orientierungsfähigkeiten erfordert, während er den Überblick über den gesamten Datensatz behalten soll, könnte es sinnvoll sein, eine VRE mit einer Visualisierung einzusetzen,

die von außen betrachtet wird. Das zweite Beispiel befasste sich mit der Bewertung verschiedener visueller Hilfsmittel zur Verbesserung der Orientierung in einer VR Umgebung. Im Gegensatz zum ersten Beispiel wurden also nicht einzelne Eigenschaften von VREs, sondern Bedingungen verglichen. In einer quantitativen Nutzerstudie wurde den Teilnehmern eine bestimmte visuelle Hilfe zur Verfügung gestellt, um verschiedene Pfadfindungs- und Navigationsaufgaben zu lösen. Unsere Ergebnisse untermauerten den Nutzen des Einsatzes von Orientierungshilfen, lieferten Richtlinien, wann welche visuelle Hilfe verwendet werden sollte, und zeigten die Vor- und Nachteile der einzelnen Techniken auf.

Die wahrscheinlich größte Einschränkung quantitativer Evaluierungen wie der hier vorgestellten besteht darin, dass bestimmte Eigenschaften in einem bestimmten Kontext (d. h. Umgebung, Technologie, Interaktionsmodalitäten) bewertet werden. In diesem Kontext ist es fast unmöglich, jeden Einflussfaktor zu bestimmen und Schlussfolgerungen über seine Auswirkungen zu ziehen. Daher sind die Ergebnisse möglicherweise nicht dieselben, wenn dasselbe Prinzip in einem anderen Kontext angewendet wird. Wir haben zum Beispiel gezeigt, dass ein höheres Maß an Immersion die Erinnerung der Teilnehmer an 3D-Streudiagramme in einer VR-Umgebung erhöht. Es ist nicht einfach, die Annahme einer erhöhten Erinnerungsleistung auf andere Arten von Visualisierungen zu übertragen, da sich zu viele Faktoren ändern. Während die allgemeinen Ideen unserer Schlussfolgerungen, wie z. B. “Immersion kann die Orientierung und das Erinnerungsvermögen verbessern”, wahrscheinlich auch für zukünftige Geräte und immersive Umgebungen gelten werden, sind Detail-Schlussfolgerungen anfälliger für technologische Fortschritte. Zukünftige Arbeiten können die Ergebnisse der vorliegenden Studien als Ausgangspunkt für neue Hypothesen nutzen, um die Schlussfolgerungen mit neuen Technologien neu zu bewerten. Darüber hinaus kann die Grundlagenforschung zu den Eigenschaften und Merkmalen immersiver Umgebungen nach und nach Richtlinien und Standards hervorbringen, die die Entwicklung effizienter und effektiver Analyseumgebungen unterstützen.

Ganzheitliche Bewertung spezifischer Anwendungen.
Das letzte Hauptkapitel behandelt die dritte Art von Bewertungsstrategien: die ganzheitliche Bewertung von IA-Techniken und -Anwendungen. Während sich die zweite Strategie auf die Auswirkungen spezifischer IA-Eigenschaften konzentrierte, wird bei dieser Art der Bewertung die allgemeine Anwendbarkeit einer Technik oder Anwendung auf eine bestimmte Gruppe von Daten und Aufgaben überprüft. Insbesondere kann dieser Ansatz bestätigen, dass bestimmte Eigenschaften, die zuvor nur in einem weit von der realen Welt entfernten Laborkontext bewertet werden konnten, auch für reale Anwendungen gelten. Wenn beispielsweise eine frühere Studie gezeigt hat, dass eine bestimmte orientierungsunterstützende Technik die Orientierung in einer Umgebung verbessern kann, die Desorientierung begünstigt, wie z. B. ein Labyrinth, könnte die Technik gezielt für den Einsatz in einer realistischeren Analyseumgebung eingesetzt werden. So könnte die Technik beispielsweise in einem Szenario zur visuellen Erkundung eines Tatorts eingesetzt werden, um die Vorteile einer verbesserten Orientierung zu nutzen. Anschließend kann diese Anwendung im Hinblick auf die gewünschten und erwarteten Auswirkungen auf die Orientierung evaluiert und bewertet werden. In diesem Kapitel werden zwei Beispiele für solche Evaluierungen angeführt (s. Abbildung 3). Die erste Anwendung ist ein An-

satz zum Vergleich mehrerer 3D-Verteilungen. Wir stellten die Hypothese auf, dass eine verbesserte Tiefenwahrnehmung und direkte Interaktionsmöglichkeiten die Leistung des Benutzers bei verschiedenen einfachen Analyseaufgaben verbessern würden. Die Ergebnisse bestätigten einen Vorteil von VR gegenüber ihrem nicht-immersiven Gegenstück für verschiedene Aufgaben. Ein besonderer Beitrag des vorgestellten Ansatzes ist das universell übertragbare Studiendesign mit einem Quervergleich von Medium und Visualisierung. Häufig werden in Evaluationen, die Bildschirme und VREs vergleichen, nur zwei Bedingungen verglichen: eine 2D-Visualisierung auf einem Bildschirm mit einer 3D-Visualisierung in VR. Das von uns vorgeschlagene Studiendesign trennt die Dimensionalität der Visualisierung vom Medium und betrachtet die beiden als unabhängige Variablen, die in einem 2x2-Kreuzvergleich bewertet werden. Obwohl dies nicht erwartet wurde, zeigten unsere Ergebnisse, dass es wichtig ist, die Auswirkungen jeder Dimension separat zu untersuchen, da die Teilnehmer teilweise in der 2D-Bedingung in VR besser abschnitten als in der 2D-Bedingung auf dem Bildschirm. Im zweiten Beispiel wurde ein immersiver Analyseansatz für die Analyse von 4D-Tatortrekonstruktionen vorgestellt. Es veranschaulicht die vielseitigen Anwendungsbereiche von Immersive Analytics, indem es intuitive Interaktionsdesigns präsentiert, verschiedene Möglichkeiten der Zusammenarbeit aufzeigt und eine hybride Analyseplattformen darstellt, die die Vorteile von bildschirmbasierten und immersiven Analyseumgebungen kombinieren. Zukünftige Forschungen sollten weiterhin neue Techniken und Anwendungen für Immersive Analytics entwickeln und sich selektiv jene Eigenschaften zunutze machen welche in früheren Evaluierungen positiv vermerkt wurden. Ähnlich wie bei den Vorzügen der von uns vorgestellten Technik ist davon auszugehen, dass es noch viele weitere Nischenanwendungen und Aufgaben gibt, die teilweise von einer Inspektion im immersiven Raum profitieren und damit besonders in hybriden Anwendungen glänzen könnten.

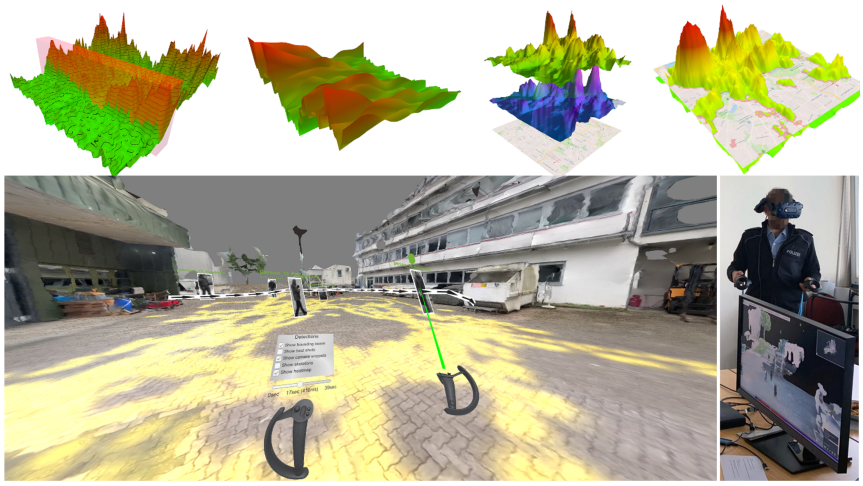


Abb. 3: Zwei Beispiele zur ganzheitlichen Bewertung immersiver Analyseumgebungen. Oben: Technik zur vergleichenden Analyse von zweidimensionalen Verteilungen welche als 3D Höhenkarten dargestellt werden. Unten: Analyseumgebung zur virtuellen Begehung eines rekonstruierten Tatortes welcher mit zusätzlichen Informationen bereichert wird.

4 Fazit

Die im Rahmen meiner Dissertation [Kr21] durchgeführte Forschung sollte der Frage auf den Grund gehen, was der Mehrwert von Immersive Analytics ist und wie er bewertet werden kann. Sicherlich kann diese Frage nicht in einer einzigen Arbeit vollständig beantwortet werden, aber ein Beitrag konnte geleistet werden. Ich habe das Bewertungsspektrum in drei Strategien unterteilt, um den Wert eines IA Szenarios zu bestimmen. Für jede Strategie habe ich mehrere Realisierungen von entsprechenden Bewertungen vorgestellt. Ich hoffe, dass die vorgestellten Ergebnisse anderen dabei helfen können, (a) neue IA-Szenarien zu bewerten, (b) zu entscheiden, ob VR eingesetzt werden soll, und (c) neue Hypothesen für künftige VR-Konfigurationen zu formulieren. Darüber hinaus hoffe ich, dass meine Forschung zum Teil dazu genutzt werden kann, fundierte Richtlinien für Immersive Analytics und für die Gestaltung von analytischen VR-Umgebungen zusammenzutragen.

Literaturverzeichnis

[Ca08] Carpendale, Sheelagh: Evaluating information visualizations. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Jgg. 4950 LNCS, S. 19–45. Springer, 2008.

[CGS69] Cohen, S.; Glaser, Barney G.; Strauss, Anselm L.: The Discovery of Grounded Theory: Strategies for Qualitative Research, Jgg. 20. Aldine, Chicago, 1969.

[Is13] Isenberg, Tobias; Isenberg, Petra; Chen, Jian; Sedlmair, Michael; Moller, Torsten: A systematic review on the practice of evaluating visualization. IEEE Transactions on Visualization and Computer Graphics, 19(12):2818–2827, 2013.

[Kr21] Kraus, Matthias: Assessing the Applicability of Virtual Reality for Data Visualization. Dissertation, Universität Konstanz, Konstanz, 2021. `http://nbn-resolving.de/urn:nbn:de:bsz:352-2-fpgbjyth97jb5`.

[La12] Lam, Heidi; Bertini, Enrico; Isenberg, Petra; Plaisant, Catherine; Lam, Heidi; Bertini, Enrico; Isenberg, Petra; Plaisant, Catherine; Seven, Sheelagh Carpendale: Seven Guiding Scenarios for Information Visualization Evaluation. Technical Report No. 2011-992-04 Department of Computer Science University of Calgary, 2012.

Matthias Kraus wurde am 12. Juni 1991 in Immenstadt im Allgäu geboren. Nachdem er 2011 sein Abitur im bayrischen Immenstadt abschloss, studierte er die folgenden Jahre Informatik an der Universität Konstanz mit einer einjährigen Unterbrechung durch einen Studienaufenthalt in den USA (University of Akron, Ohio). Nach dem erfolgreich abgeschlossenen Masterstudium im Jahr 2017 began er seine Promotion im Bereich Datenanalyse und Visualisierung (LS Prof. Dr. Daniel Keim) in Konstanz welche nach vier Jahren mit Auszeichnung (summa cum laude) abgeschlossen wurde. Der Fokus seiner Promotion lag auf der Untersuchung verschiedener Einsatzmöglichkeiten für Immersion im Bereich der Datenanalyse.

Das Re-Engineering variantenreicher Systeme verstehen: Eine empirische Arbeit über Kosten, Wissen, Nachvollziehbarkeit und Methoden[1]

Jacob Krüger[2]

Abstract: Durch variierende Anforderungen existieren die meisten Softwaresysteme in verschiedenen Varianten. Entwickler beginnen üblicherweise durch Klonen und Anpassen diese wiederzuverwenden, bis Wartungsprobleme zum Re-Engineering einer Softwareplattform führen. Obwohl dies das verbreitetste Szenario ist, wurde es in der Forschung nur unzureichend untersucht. Im Rahmen der Dissertation wurden vier Kernfaktoren empirisch analysiert, was zu folgenden, stark zusammengefassten, Ergebnissen führt: Entwickler sollten versuchen die Wiederverwendung in Richtung einer Softwareplattform zu systematisieren. Wissen über Features und deren Sourcecode ist essentiell, weshalb Featurecode proaktiv dokumentiert werden sollte, da ansonsten Wissen aufwändig wiedergewonnen werden muss. Ein aktualisiertes Prozessmodell mit zugehörigen Richtlinien hilft Organisationen bei der Durchführung von Re-Engineering Projekten und zeigt neue Forschungsrichtungen auf. Die Ergebnisse stellen eine Synthese und Erweiterung des existierenden Wissensstandes zum (Re-)Engineering variantenreicher Systeme sowie weiterer grundsätzlicher Problemstellungen dar, durch die viele etablierte Annahmen mit verlässlichen und aktuellen Daten bestätigt aber auch einige widerlegt werden.

1 Einleitung

Moderne Softwaresysteme in allen Domänen existieren in verschiedenen Varianten um unterschiedliche Anforderungen von Kunden, bestimmter Hardware oder anderen Beschränkungen zu erfüllen. Solche variantenreichen Systeme sind in verschiedenen Ausprägungen sowohl in der Industrie als auch im Open-Source Bereich weit verbreitet, wie beispielsweise bei Betriebssystemen (z.B. Linux Kernel und Distributionen, Windows), eingebetteten Systemen (z.B. 3D Drucker, Roboter, Kameras, Steuerungseinheiten), Programmiersprachen (z.B. GCC, Python), Android Apps, Datenbanken (z.B. Oracle Berkeley DB) oder Autos (z.B. Volve, Opel, Rolls Royce) [Be20; Fo16; KB20b; Ku18; LSR07; SSW15; YGM06]. Da sich die entstehenden Varianten sehr ähneln, ist es eine wichtige strategische Entscheidung für jede Organisation, wie sie die Wiederverwendung bereits existierender Softwareartefakte (z.B. Code, Modelle, Anforderungen) organisiert.

[1] Englischer Titel der Dissertation [Kr21a]: „Understanding the Re-Engineering of Variant-Rich Systems: An Empirical Work on Economics, Knowledge, Traceability, and Practices“

[2] Eindhoven University of Technology, Department of Mathematics and Computer Science, Groene Loper 3, 5612 AE Eindhoven, The Netherlands, j.kruger@tue.nl

Wiederverwendung ist ein Kernprinzip des Software Engineering um Entwicklungs- und Wartungskosten zu reduzieren, eine neue Variante schneller zu liefern und die Softwarequalität zu erhöhen [KB20b; LSR07; St84]. Es existiert eine Vielzahl an Techniken zur Wiederverwendung von Softwareartefakten, die in zwei Strategien unterteilt werden können:

Klonen bezeichnet die Strategie eine existierende Variante (oder größere Teile von dieser) zu Kopieren und an geänderte Anforderungen anzupassen [SSW15]. Die Vorteile dieser Strategie liegen darin, dass sie ohne weitere Vorbereitung umsetzbar und dadurch ohne größere Investitionen nutzbar ist, sowie von vielen Versionskontrollsystemen unterstützt wird [KB20b; SSW15]. Aus diesem Grund beginnen viele Organisationen mit dieser Strategie, bis die Wartung der unabhängigen Klone zu aufwendig wird, beispielsweise weil Änderungen propagiert werden müssen oder die Entwickler den Überblick verlieren [Be20; Ku18; YGM06].

Softwareplattformen hingegen erfordern, dass die Organisation ein systematisch wiederverwendbares Portfolio an Softwareartefakten entwickelt, das es erlaubt Varianten zu konfigurieren und automatisch zu generieren. Diese Strategie baut typischerweise auf Produktlinienkonzepten auf (z.B. Variabilitätsmechanismen, Feature-Modelle) und organisiert existierende Artefakte entlang (konfigurierbarer) Funktionalitäten der Varianten—welche als Features bezeichnet werden [LSR07]. Im Gegensatz zum Klonen erfordert eine Softwareplattform hohe Investitionen in technische (z.B. Plattformarchitektur) und nicht-technische (z.B. Prozesse) Aspekte [KB20b; Li21; LSR07]. Allerdings versprechen Softwareplattformen verglichen mit dem Klonen erhebliche Einsparungen bei der Entwicklung und Wartung von Varianten, erhöhte Softwarequalität und eine schnellere Auslieferung.

Da die meisten Organisationen mit Klonen beginnen, ist das Re-Engineering von geklonten Varianten in eine Softwareplattform eines der häufigsten praktischen Re-Engineering Szenarien mit hohen Risiken und Kosten [Be13; Be20; Kr16; Kr19a; Ku18; YGM06]. Beispielsweise berichten Fogdal et al. [Fo16], dass Danfoss für das Re-Engineering von geklonten Varianten in eine Softwareplattform 10 Monate geplant hatte (inklusive Sourcecode, Tools, Feature-Modell und weiteren Artefakten). Stattdessen benötigte das Unternehmen alleine 36 Monate um 80 % des Sourcecodes zu überführen.

Im Rahmen der Dissertation [Kr21a] wurden vier Faktoren des Re-Engineerings variantenreicher Systeme erforscht und als Forschungsziele (FZ) definiert, die für das Verständnis und damit die Planbarkeit solcher Projekte essentiell sind. Aktuelle Literaturanalysen, Umfragen unter Experten und Erfahrungsberichte zeigen, dass diese Faktoren unzureichend verstanden sind und keine systematisch erhobenen Daten existieren. Dementsprechend basieren viele Entscheidungen bezüglich des (Re-)Engineerings variantenreicher Systeme lediglich auf einzelnen Erfahrungen, Spekulationen und Bauchgefühlen, trotz Jahrzehnten an Forschung. Abb. 1 zeigt einen groben Überblick über die folgenden vier Faktoren, ihre Beziehungen zueinander, was im Rahmen der Dissertation analysiert wurde sowie

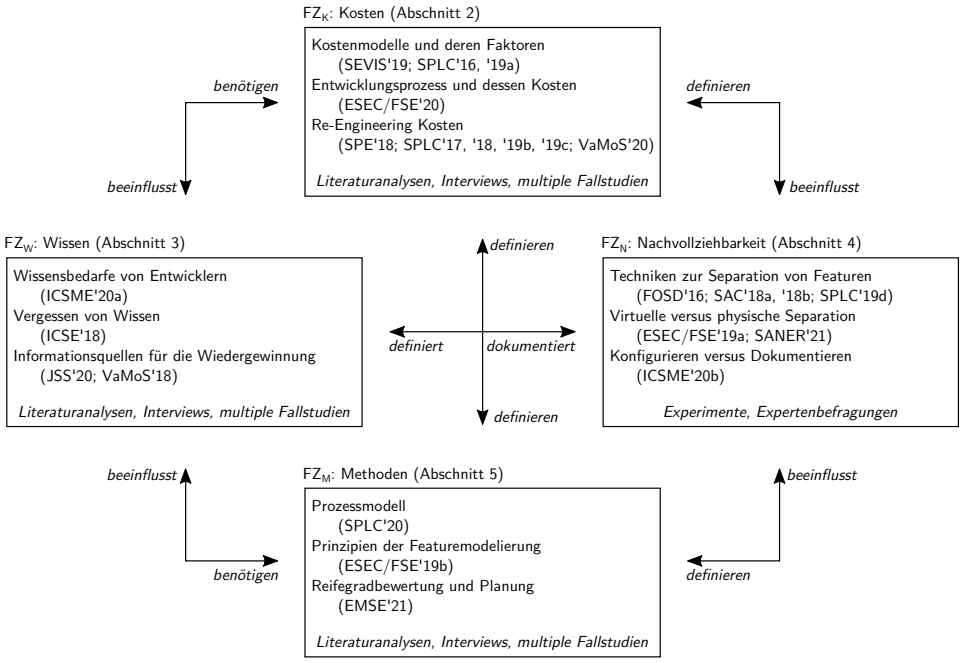


Abb. 1: Grobüberblick über die Forschungsziele, ihre Beziehungen (Pfeile) und zugehörigen Beiträge mit den wichtigsten Publikationen (in Klammern) sowie Forschungsmethoden (kursiv).

die wichtigsten Publikationen (in Klammern) und Forschungsmethoden (kursiv)—eine detailliertere Beschreibung folgt in den zugehörigen Abschnitten:

FZ_K – Kosten (Abschnitt 2): Die Kosten des (Re-)Engineerings variantenreicher Systeme zu verstehen hilft, strategische Entscheidungen bezüglich der zu nutzenden Strategie zu treffen, das Projektmanagement zu unterstützen (FZ_W) und besonders anspruchsvolle Aktivitäten zu identifizieren sowie zu unterstützen (FZ_N, FZ_M).

FZ_W – Wissen (Abschnitt 3): Für das erfolgreiche (Re-)Engineering variantenreicher Systeme benötigen die involvierten Entwickler Wissen, was den größten Kostenfaktor darstellt (FZ_K). Zu verstehen, welches Wissen dabei (nicht mehr) verfügbar ist und von wo es zurückgewonnen werden kann hilft, Entwickler beim Verstehen eines Systems zu unterstützen (FZ_N) und in Entwicklungsprozesse einzubinden (FZ_M).

FZ_N – Nachvollziehbarkeit (Abschnitt 4): Das teuerste (FZ_K) Wissen (FZ_W) im Bezug auf das (Re-)Engineering variantenreicher Systeme sind die genauen Stellen im Sourcecode die ein bestimmtes Feature implementieren („Feature Locations“). Eine geeignete Technik (FZ_M) für die expliziete Nachvollziehbarkeit von Featuren proaktiv einzuführen ist essentiell um später Probleme und Kosten zu vermeiden.

FZ_M – Methoden (Abschnitt 5): Existierende Richtlinien für die Entwicklung variantenreicher Systeme sind seit Jahrzehnten nicht systematisch aktualisiert worden. Ein Prozessmodell und Analysemethoden die moderne Software Engineering Techniken (FZ_N) berücksichtigen helfen bei der Planung und beim Monitoring von (Re-)Engineering Projekten (FZ_K) sowie der Organisation von Wissen (FZ_W).

Um systematisch Daten bezüglich dieser Faktoren zu erheben, wurde auf Methoden zur Kreation und Verifikation von Wissen im Rahmen des Evidence-Based Software Engineerings aufgebaut [SSS08], beispielsweise Literaturanalysen, Experimente und multiple Fallstudien. Durch das bessere Verständnis und die verlässlicheren Daten, die im Rahmen der Dissertation bezüglich der vier Faktoren entstanden sind, können Organisationen verlässlicher Entscheidungen treffen und Projekte managen, sowie Forscher praxisrelevante Ansätze entwickeln und einige etablierte Missverständnisse auflösen. Dies ist durch Kollaborationen (z.B. mit Axis AB, pure-systems GmbH) während der Forschungsarbeiten sichergestellt und durch spätere Diskussionen der Ergebnisse (z.B. mit Danfoss SE, Robert Bosch GmbH) bestätigt worden. Zudem gehen einige der Einsichten, beispielsweise zu Wissen, Nachvollziehbarkeit und dem verbindenden Thema Programmverständnis, über den Bereich des (Re-)Engineerings variantenreicher Systeme hinaus und sind für das Software Engineering im Allgemeinen von großer Bedeutung.

2 FZ_K: Kosten variantenreicher Systeme

Die Kosten und Nutzen beider Wiederverwendungsstrategien gegeneinander abzuwägen ist essentiell damit eine Organisation eine verlässliche Entscheidung treffen und Projekte planen kann. Zudem hängt von dieser Entscheidung ab, was für Wissen (z.B. einzelne Variante versus gemeinsame Softwareplattform), Nachvollziehbarkeit (z.B. Kunden zu geklonten Varianten oder zu Features der Softwareplattform) und Methoden (z.B. kontinuierliche Integration bei einer Softwareplattform) benötigt werden. Existierende Kostenmodelle und Analysemethoden basieren oft auf einzelnen Erfahrungen anstatt von systematisch erhobenen Daten, weshalb die meisten Entscheidungen auf Bauchgefühlen beruhen [KB20a; KB20b; Kr16; Li21]. Verschiedene Literaturanalysen [As17; LC13] und Expertenbefragungen [Be20; GMA12; Ra19] verdeutlichen, dass es sich dabei um ein langanhaltendes und fundamentales Problem handelt—was insbesondere in der Komplexität von Kostenanalysen im Software Engineering begründet liegt [Bo84; JB09].

Im Rahmen der Dissertation wurden existierende Kostenmodelle für Softwareplattformen und deren Faktoren analysiert [KBL19; Kr16]. Basierend auf dieser Analyse wurden Forschungsarbeiten gesammelt und Interviews mit Entwicklern bei Axis AB durchgeführt um quantitative als auch qualitative Daten zur Wiederverwendung in über 100 Organisationen zu sammeln [KB20b]. Der Vergleich zwischen beiden Wiederverwendungsstrategien hat viele etablierte Annahmen zu den Kosten bestätigt (z.B. Features einer Softwareplattform sind teurer als die von Klonen), aber auch einige widerlegt (z.B. Änderungen propagieren ist

teurer in Klonen). Zuletzt wurden multiple Fallstudien durchgeführt um die Probleme und Kosten des Re-Engineerings von Klonen zu einer Softwareplattform zu verstehen [De19; KB20a; Kr17; Kr18c; Ku18]. Aus allen drei Teilbereichen ging hervor, dass eine Plattform deutliche Kostenersparnisse gegenüber dem Klonen bietet, aber auch dass dies primär vom Wissen der Entwickler und der Softwarequalität abhängt. Auf einer sehr hohen Abstraktionsebene ist die Kerneinsicht zu diesem Faktor:

FZ_K: Kosten

Die Wiederverwendung von Softwareartefakten in Richtung einer Softwareplattform zu systematisieren ist die ökonomisch sinnvollste Strategie.

3 FZ_W: Das Wissen der Entwickler

Obwohl eine Vielzahl von Faktoren (z.B. Systemgröße, Methoden) die Kosten des (Re-) Engineerings variantenreicher Systeme beeinflussen, hat sich in den Analysen das (verbliebene) Wissen der Entwickler als besonders wichtig herausgestellt. Vor allem zu identifizieren, welcher Sourcecode welches Feature implementiert und das zugehörige Programmverständnis wiederzuerlangen sind Herausforderungen in diesem Bereich. Beides benötigt hohen manuellen Aufwand, da eine Automatisierung bestenfalls unzuverlässig möglich ist und meist keine Methoden zur Nachvollziehbarkeit verwendet oder aktualisiert wurden [BMW93; KBL19; Ra18b; Wa13]. Andere Studien mit Entwicklern zeigen, dass diese weiteres Wissen bezüglich der Features benötigen und dies wiederum nicht umfangreich genug untersucht worden ist [Be20; GMA12; Ra18a; Ta10].

Im Rahmen der Dissertation wurde eine Literaturanalyse durchgeführt um darauf aufbauend Entwickler zu interviewen um zu verstehen, welches Wissen ihnen wichtig ist und welches sie sich merken [KH20]. Hieraus ging hervor, dass Entwickler sich vorrangig abstraktes Wissen (z.B. welche Features existieren) merken, aber nicht Details (z.B. die zugehörigen Stellen im Sourcecode). Mit einer weiteren Befragung von Entwicklern wurde untersucht, wie sich deren verbliebenes Programmverständnis zu einen Stück Sourcecode über die Zeit entwickelt [Kr18d]. Da Entwickler ihr Wissen in verschiedenen Bereichen aber insbesondere bezüglich ihres Programmverständnisses wiedererlangen müssen, wurde in einer multiplen Fallstudie untersucht, welche Informationsquellen sie dafür nutzen können [Kr18a; Kr19c]. Aus diesen drei Teilbereichen lassen sich verschiedene Empfehlungen für Organisationen ableiten, beispielsweise, dass mit dem System erfahrene Entwickler neuen Entwicklern die abstrakte Struktur vermitteln können. Auf einer sehr hohen Abstraktionsebene ist die Kerneinsicht zu diesem Faktor:

FZ_W: Wissen

Wissen bezüglich der Features eines variantenreichen Systems und deren zugehöriger Sourcecode sollten dokumentiert werden um teure Wiedergewinnung zu vermeiden.

4 FZ_N: Nachvollziehbarkeit von Wissen

Nachvollziehbarkeit zielt darauf ab, Wissen explizit zu dokumentieren und über verschiedene Artefakte verfolgen zu können. Da die vorigen Ergebnisse gezeigt haben, dass insbesondere das Wissen welcher Sourcecode ein bestimmtes Feature implementiert wichtig ist und die Wiedergewinnung Kosten verursacht, lag der weitere Fokus der Dissertation auf der Nachvollziehbarkeit von Featuren. In variantenreichen Systemen werden meist nur optionale Features zu einem gewissen Grad durch den Variabilitätsmechanismus im Sourcecode dokumentiert, der aber zusätzliche Komplexität hinzufügt und oftmals nicht eindeutig ist (z.B. können Präprozessor Direktiven beliebig viele Features komplex miteinander verbinden). Methoden zur Nachvollziehbarkeit von Featuren einzuführen ist herausfordernd und es fehlt an empirischen Daten, die die Wirksamkeit verschiedener Methoden untermauern sowie von Variabilitätsmechanismen abgrenzen [As17; Be20; LC13; Va17].

Im Rahmen der Dissertation wurden die Dimensionen der Nachvollziehbarkeit von Featuren basierend auf Codeanalysen und Expertenbefragungen erforscht [Kr18b; LKL19]. Darauf aufbauend entstanden zwei Experimente mit verschiedenen Fokussen: Erstens, ob Features durch Annotationen oder durch die physische Trennung (z.B. als Module) das Programmverständnis besser unterstützen [Kr19b; Kr21b]. Die Ergebnisse deuten auf Annotationen im Sourcecode als geeignetere Methode hin, da sie nicht zu Indirektionen zwischen interagierenden Teilen des Systems führen. Zweitens, welchen Einfluss die Konfigurierbarkeit von Featureannotationen auf das Programmverständnis hat [Fe20]. Neben einer überraschenden Diskrepanz zwischen der Wahrnehmung und der objektiven Leistung der Teilnehmer des Experiments, deuten die Ergebnisse darauf hin, dass die Nachvollziehbarkeit von Featuren besser über Annotationen sichergestellt wird die nicht konfigurierbar sind. Dementsprechend lässt sich aus diesen drei Teilen eine klare Empfehlung zur Implementierung der Nachvollziehbarkeit von Featuren ableiten. Auf einer sehr hohen Abstraktionsebene ist die Kerneinsicht zu diesem Faktor:

FZ_N: Nachvollziehbarkeit

Features sollten im Sourcecode mit nicht konfigurierbaren Annotationen dokumentiert werden um durch die Nachvollziehbarkeit das Programmverständnis zu vereinfachen.

5 FZ_M: Methoden des (Re-)Engineerings

Die Bearbeitung der vorigen Forschungsziele war eng gekoppelt an bestimmte Prozesse, Aktivitäten und Methoden für das (Re-)Engineering variantenreicher Systeme (z.B. entstehen Kosten durch Aktivitäten). Dabei wurde klar, dass die existierenden Prozessmodelle und viele Methoden für solche Systeme seit Jahrzehnten nicht aktualisiert worden sind—was durch weitere Literaturanalysen [Ra18a] und industrielle Studien [Be20] ebenfalls untermauert wird. Ein aktualisiertes Prozessmodell und neue Methoden die die vorigen Ergebnisse berücksichtigen bieten Organisationen konkrete Unterstützung bei ihren Projekten und vermitteln Forschern den aktuellen Stand der Praxis.

Im Rahmen der Dissertation wurden die vorigen Ergebnisse synthetisiert und kontextualisiert indem sie, erweitert um eine zusätzliche Literaturanalyse, für die Entwicklung des neuen Prozessmodells promote-pl genutzt wurden [KMB20]. Dies inkorporiert die Erfahrungen mit Mischformen der variantenreichen Systeme die in anderen Studien auftraten und beinhaltet besonders Wissens- und Nachvollziehbarkeitsaspekte. Für die initiale Phase eines (Re-)Engineeringprozesses in dem der Umfang und die Kosten des Systems definiert werden, entstanden zudem Prinzipien für die Feature Modellierung aus einer Literaturanalyse und Experteninterviews um damit verbundene Fallstricke zu vermeiden [Ne19]. Zudem ist es wichtig, dass ein variantenreiches System vor und während seiner Lebenszeit nach verschiedenen Dimensionen (z.B. Kosten, Methoden) analysiert wird. Für diesen Zweck existieren Reifegradmodelle, von denen eines im Rahmen einer multiplen Fallstudie in einem Unternehmen angewandt wurde [Li21]. Daraus entstanden Empfehlungen zur Anwendung, den Vorteilen und den Problemen dieses Modells. Diese drei Teile bilden ein Rahmenwerk für Organisationen, ihre (Re-)Engineeringprojekte unter Berücksichtigung aktueller Standards effizient zu planen und durchzuführen. Auf einer sehr hohen Abstraktionsebene ist die Kerneinsicht zu diesem Faktor:

FZ_M: Methoden

Das Prozessmodell und die Methoden bilden ein Rahmenwerk um das (Re-)Engineering eines variantenreichen Systems durchzuführen und den Nutzen zu maximieren.

6 Zusammenfassung

Die Ergebnisse der Dissertation tragen dazu bei, das Re-Engineering variantenreicher Systeme besser zu verstehen. Dabei sind unter Anderem verlässliche ökonomische Daten (FZ_K), ein besseres Verständnis für Wissensbedarfe (FZ_W), Empfehlungen für die Implementierung von Nachvollziehbarkeit (FZ_N) und Methoden für das Projektmanagement (FZ_M) entstanden. Diese Beiträge erweitern den wissenschaftlichen Wissensstand beträchtlich durch belastbare empirische Einsichten und werden bereits in der Praxis genutzt. Allerdings kann diese Zusammenfassung lediglich einen sehr abstrakten Überblick über die einzelnen Forschungsziele der Dissertation geben, die bedeutend tiefer gehen.

Literatur

[As17] Assunção, W. K. G.; Lopez-Herrejon, R. E.; Linsbauer, L.; Vergilio, S. R.; Egyed, A.: Reengineering Legacy Applications into Software Product Lines: A Systematic Mapping. Empirical Software Engineering 22/6, 2017.

[Be13] Berger, T.; She, S.; Lotufo, R.; Wąsowski, A.; Czarnecki, K.: A Study of Variability Models and Languages in the Systems Software Domain. IEEE Transactions on Software Engineering 39/12, 2013.

[Be20] Berger, T.; Steghöfer, J.-P.; Ziadi, T.; Robin, J.; Martinez, J.: The State of Adoption and the Challenges of Systematic Variability Management in Industry. Empirical Software Engineering 25/, 2020.

[BMW93] Biggerstaff, T. J.; Mitbander, B. G.; Webster, D. E.: The Concept Assignment Problem in Program Understanding. In: WCRE. IEEE, 1993.

[Bo84] Boehm, B. W.: Software Engineering Economics. IEEE Transactions on Software Engineering SE-10/1, 1984.

[De19] Debbiche, J.; Lignell, O.; Krüger, J.; Berger, T.: Migrating Java-Based Apo-Games into a Composition-Based Software Product Line. In: SPLC. ACM, 2019.

[Fe20] Fenske, W.; Krüger, J.; Kanyshkova, M.; Schulze, S.: #ifdef Directives and Program Comprehension: The Dilemma between Correctness and Preference. In: ICSME. IEEE, 2020.

[Fo16] Fogdal, T. S.; Scherrebeck, H.; Kuusela, J.; Becker, M.; Zhang, B.: Ten Years of Product Line Engineering at Danfoss: Lessons Learned and Way Ahead. In: SPLC. ACM, 2016.

[GMA12] Ghanam, Y.; Maurer, F.; Abrahamsson, P.: Making the Leap to a Software Platform Strategy: Issues and Challenges. Information and Software Technology 54/9, 2012.

[JB09] Jørgensen, M.; Boehm, B. W.: Software Development Effort Estimation: Formal Models or Expert Judgment? IEEE Software 26/2, 2009.

[KB20a] Krüger, J.; Berger, T.: Activities and Costs of Re-Engineering Cloned Variants Into an Integrated Platform. In: VaMoS. ACM, 2020.

[KB20b] Krüger, J.; Berger, T.: An Empirical Analysis of the Costs of Clone- and Platform-Oriented Software Reuse. In: ESEC/FSE. ACM, 2020.

[KBL19] Krüger, J.; Berger, T.; Leich, T.: Features and How to Find Them - A Survey of Manual Feature Location. In: Software Engineering for Variability Intensive Systems. CRC Press, 2019.

[KH20] Krüger, J.; Hebig, R.: What Developers (Care to) Recall: An Interview Survey on Smaller Systems. In: ICSME. IEEE, 2020.

[KMB20] Krüger, J.; Mahmood, W.; Berger, T.: Promote-pl: A Round-Trip Engineering Process Model for Adopting and Evolving Product Lines. In: SPLC. ACM, 2020.

[Kr16] Krüger, J.; Fenske, W.; Meinicke, J.; Leich, T.; Saake, G.: Extracting Software Product Lines: A Cost Estimation Perspective. In: SPLC. ACM, 2016.

[Kr17] Krüger, J.; Nell, L.; Fenske, W.; Saake, G.; Leich, T.: Finding Lost Features in Cloned Systems. In: SPLC. ACM, 2017.

[Kr18a] Krüger, J.; Gu, W.; Shen, H.; Mukelabai, M.; Hebig, R.; Berger, T.: Towards a Better Understanding of Software Features and Their Characteristics. In: VaMoS. ACM, 2018.

[Kr18b] Krüger, J.; Ludwig, K.; Zimmermann, B.; Leich, T.: Physical Separation of Features: A Survey with CPP Developers. In: SAC. ACM, 2018.

[Kr18c] Krüger, J.; Pinnecke, M.; Kenner, A.; Kruczek, C.; Benduhn, F.; Leich, T.; Saake, G.: Composing Annotations Without Regret? Practical Experiences Using FeatureC. Software: Practice and Experience 48/3, 2018.

[Kr18d] Krüger, J.; Wiemann, J.; Fenske, W.; Saake, G.; Leich, T.: Do You Remember This Source Code? In: ICSE. ACM, 2018.

[Kr19a] Krüger, J.: Are You Talking about Software Product Lines? An Analysis of Developer Communities. In: VaMoS. ACM, 2019.

[Kr19b] Krüger, J.; Çalıklı, G.; Berger, T.; Leich, T.; Saake, G.: Effects of Explicit Feature Traceability on Program Comprehension. In: ESEC/FSE. ACM, 2019.

[Kr19c] Krüger, J.; Mukelabai, M.; Gu, W.; Shen, H.; Hebig, R.; Berger, T.: Where is My Feature and What is it About? A Case Study on Recovering Feature Facets. Journal of Systems and Software 152/, 2019.

[Kr21a] Krüger, J.: Understanding the Re-Engineering of Variant-Rich Systems: An Empirical Work on Economics, Knowledge, Traceability, and Practices, Diss., Otto-von-Guericke University Magdeburg, 2021.

[Kr21b] Krüger, J.; Çalıklı Gül, G.; Berger, T.; Leich, T.: How Explicit Feature Traces Did Not Impact Developers' Memory. In: SANER. IEEE, 2021.

[Ku18] Kuiter, E.; Krüger, J.; Krieter, S.; Leich, T.; Saake, G.: Getting Rid of Clone-And-Own: Moving to a Software Product Line for Temperature Monitoring. In: SPLC. ACM, 2018.

[LC13] Laguna, M. A.; Crespo, Y.: A Systematic Mapping Study on Software Product Line Evolution: From Legacy System Reengineering to Product Line Refactoring. Science of Computer Programming 78/8, 2013.

[Li21] Lindohf, R.; Krüger, J.; Herzog, E.; Berger, T.: Software Product-Line Evaluation in the Large. Empirical Software Engineering 26/30, 2021.

[LKL19] Ludwig, K.; Krüger, J.; Leich, T.: Covert and Phantom Features in Annotations: Do They Impact Variability Analysis? In: SPLC. ACM, 2019.

[LSR07] van der Linden, F. J.; Schmid, K.; Rommes, E.: Software Product Lines in Action. Springer, 2007.

[Ne19] Nešić, D.; Krüger, J.; Stănciulescu, S.; Berger, T.: Principles of Feature Modeling. In: ESEC/FSE. ACM, 2019.

[Ra18a] Rabiser, R.; Schmid, K.; Becker, M.; Botterweck, G.; Galster, M.; Groher, I.; Weyns, D.: A Study and Comparison of Industrial vs. Academic Software Product Line Research Published at SPLC. In: SPLC. ACM, 2018.

[Ra18b] Razzaq, A.; Wasala, A.; Exton, C.; Buckley, J.: The State of Empirical Evaluation in Static Feature Location. ACM Transactions on Software Engineering and Methodology 28/1, 2018.

[Ra19] Rabiser, R.; Schmid, K.; Becker, M.; Botterweck, G.; Galster, M.; Groher, I.; Weyns, D.: Industrial and Academic Software Product Line Research at SPLC: Perceptions of the Community. In: SPLC. ACM, 2019.

[SSS08] Shull, F.; Singer, J.; Sjøberg, D. I. K., Hrsg.: Guide to Advanced Empirical Software Engineering. Springer, 2008.

[SSW15] Stănciulescu, S.; Schulze, S.; Wąsowski, A.: Forked and Integrated Variants in an Open-Source Firmware Project. In: ICSME. IEEE, 2015.

[St84] Standish, T. A.: An Essay on Software Reuse. IEEE Transactions on Software Engineering SE-10/5, 1984.

[Ta10] Tang, A.; Couwenberg, W.; Scheppink, E.; de Burgh, N. A.; Deelstra, S.; van Vliet, H.: SPL Migration Tensions: An Industry Experience. In: KOPLE. ACM, 2010.

[Va17] Vale, T.; de Almeida, E. S.; Alves, V. R.; Kulesza, U.; Niu, N.; de Lima, R.: Software Product Lines Traceability: A Systematic Mapping Study. Information and Software Technology 84/, 2017.

[Wa13] Wang, J.; Peng, X.; Xing, Z.; Zhao, W.: How Developers Perform Feature Location Tasks: A Human-Centric and Process-Oriented Exploratory Study. Journal of Software: Evolution and Process 25/11, 2013.

[YGM06] Yoshimura, K.; Ganesan, D.; Muthig, D.: Assessing Merge Potential of Existing Engine Control Systems into a Product Line. In: SEAS. ACM, 2006.

Jacob Krüger ist Assistant Professor für Software Engineering an der Technischen Universität Eindhoven in den Niederlanden. Er hat zuvor an der Ruhr-Universität Bochum als Akademischer Rat gearbeitet nachdem er an der Otto-von-Guericke Universität Magdeburg seinen Masterabschluss in Wirtschaftsinformatik (2016) sowie seinen Doktor in der Arbeitsgruppe Datenbanken und Software Engineering (2021) erlangte. Dabei arbeitete er als wissenschaftlicher Mitarbeiter an dem DFG-Projekt EXPLANT, zuerst an der Fachhochschule Harz und danach an der Otto-von-Guericke Universität. Während seiner Promotion hat er einen einjährigen Forschungsaufenthalt an der Technischen Hochschule Chalmers | Universität von Göteborg in Schweden sowie einen dreimonatigen Forschungsaufenthalt an der Universität von Toronto in Kanada wahrgenommen. Seine Forschung wurde durch Preise, Stipendien und eigene Projekte ausgezeichnet beziehungsweise unterstützt und in führenden Konferenzen (z.B. ICSE, ESEC/FSE, ICSME) sowie Journalen (z.B. EMSE, JSS) publiziert. Er fokussiert sich auf menschliche Faktoren während der Softwareevolution, insbesondere auf Kosten, Programmverständnis und variantenreiche Systeme.

Sprachrepräsentationen für Rechnerische Argumentation[1]

Anne Lauscher[2]

Abstract: Rechnerische Argumentation gilt als eines der komplexesten Anwendungsfelder der Künstlichen Intelligenz. Wie frühere Forschung zeigte, erfordert sie daher hochentwickelte numerische Sprachrepräsentationen. Obwohl das Lernen solcher Repräsentationen zu einem Kernforschungsfeld in der Verarbeitung natürlicher Sprache zählt, gibt es bis heute keine systematische Forschung, die Sprachrepräsentationen für rechnerische Argumentation untersucht. Wir adressieren diese Forschungslücke indem wir fünf Herausforderungen an der Schnittstelle der beiden Areale ableiten und in anwendungsbezogenen Fallstudien bearbeiten: Externes Wissen, Domänenwissen, geteiltes Wissen zwischen Aufgaben, Mehrsprachigkeit und ethische Überlegungen. In diesem Zuge schlagen wir neue Methoden (z.B. zur effizienten Wissensinjizierung), Ressourcen (z.B. neue Annotationsebenen zur Argumentationsstruktur) und Maße (z.B. zur Bestimmung stereotypischer Verzerrungen) vor. Unsere Beiträge werden effektive, effiziente, inklusive und faire Argumentationstechnologien voranbringen.

1 Einleitung

Argumentation ist eines der wohl aufregendsten Phänomene in der Benutzung natürlicher Sprache und wurde bereits im antiken Griechenland als Mittel zur politischen Meinungsbildung diskutiert [z.B. Ar06]. Auch ihre computergestützte Analyse, die unter den Begriff *Computational Argumentation* (CA) fällt, ist in einer Vielfalt von Anwendungen nützlich. Sie kann dabei helfen, gezieltes Feedback zur Qualität argumentativer Texte zu geben [z.B. Gr20] und automatisch wissenschaftliche Publikationen zusammenzufassen [z.B. AR11].

Wie in allen Bereichen in Natural Language Processing (NLP) muss der Text, der analysiert werden soll, den CA-Modellen in Form numerischer Features eingegeben werden. Repräsentationen zu finden, die die Semantik eines Texts adäquat reflektieren, wird als eine der Kernfragestellungen in NLP betrachtet. In diesem Kontext erzielte Forschung, die vortrainierte statische [z.B. Mi13] und kontextualisierte Embedding-Methoden [z.B. De19] einsetzt, state-of-the-art Ergebnisse in einer Reihe von grundlegenden Textverstehensaufgaben [Wa19]. Vorhergegangene Arbeiten haben jedoch bereits die spezifische Schwierigkeit von CA-Szenarien erkannt: Während es beispielsweise im verwandten Bereich des Opinion Minings darum geht, zu erkennen, *was* User*innen denken, geht CA einen Schritt weiter und versucht zu verstehen, *warum* sie so denken [HG16]. Die Komplexität dieser Frage nach dem *warum* mit den dahinterliegenden logischen Schlussfolgerungsprozessen, dem relevanten themenspezifischen Wissen und den rhetorischen Aspekten im Rahmen des bestimmten

[1] Englischer Titel der Dissertation: "Language Representations for Computational Argumentation"

[2] Universität Mannheim, Fakultät für Wirtschaftsinformatik und Wirtschaftsmathematik, B6 26, 68159 Mannheim, Deutschland, anne.lauscher@web.de

soziokulturellen Kontextes einer Debatte übersteigt das Verstehen des *was* in vielerlei Hinsicht und erfordert neue NLP-Methoden. Moens [Mo18] nennt in diesem Kontext die numerischen Sprachrepräsentationen als einen Hauptengpass. Dennoch gibt es nur wenige Forschungsanstrengungen, die sich gezielt auf die Schnittstelle von Sprachrepräsentationen und CA beziehen. Wir adressieren diese Forschungslücke und erkennen die spezifische Wichtigkeit von Forschung am Zusammenspiel zwischen CA und Sprachrepräsentationen an. Dazu (1) identifizieren wir zunächst eine Serie von fünf Herausforderungen basierend auf inhärenten Charakteristika von Argumentation ((**C1**) Externes Wissen, (**C2**) Domänenwissen, (**C3**) Geteiltes Wissen zwischen Aufgaben, (**C4**) Multilingualität und (**C5**) Ethische Aspekte) und (2) präsentieren neue Analysen, Maßzahlen, Textkorpora und Methoden, um jedes der Probleme zu adressieren. Unsere vielfältigen Beiträge lassen sich zum einen dem CA-Bereich und zum anderen dem Repräsentionslernen zuordnen. Methodische Beiträge zum Repräsentationslernen werden anhand argumentativer Anwendungen demonstriert.

Forschungsbeiträge. Wir präsentieren drei neue textuelle Ressourcen, z.B. das erste Korpus englischsprachiger wissenschaftlicher Publikationen, das mit feinkörnigen argumentativen Strukturen annotiert ist, und das größte englischsprachige Multidomänen-Korpus, welches mit theoriebasierten Argumentationsqualitätsbewertungen annotiert ist. Außerdem stellen wir neue Analysewerkzeuge und Maßzahlen und neue Anpassungen dieser vor. Ein Beispiel hierfür ist *DebIE*, ein neues Framework für die implizite und explizite Bewertung stereotyper Verzerrungen in distributionellen Wortvektorräumen. In diesem Zuge präsentieren wir auch den neuen Bias Analogy Test, der das Vorhandensein einer expliziten stereotypischen Verzerrung in statischen Wortvektorräumen prüft und dessen Ausmaß quantifiziert. Die neu vorgestellten Ressourcen und Maßzahlen erlauben es uns, eine Vielzahl von Analysen durchzuführen. So sind wir die ersten, die unfaire stereotype Verzerrungen in distributionellen Wortvektorräumen über eine Vielzahl von Sprachen und in sprachübergreifenden Repräsentationsräumen quantifizieren und dabei andere relevante Faktoren wie die Repräsentationsmethode berücksichtigen. Als weiteres Beispiel analysieren wir im Kontext von Mehrsprachigkeit den aktuellen Stand der Forschung beim sprachübergreifenden Transfer, indem wir den Performanzverlust beim Transfer quantifizieren und die dabei relevanten Faktoren bestimmen. Zuletzt präsentieren wir mehrere neue Methoden, wie zum Beispiel das neue Few-Shot Target-Language Fine-Tuning, welches die festgestellten Einschränkungen des sprachübergreifenden Transfers effizient verringert. Des Weiteren stellen wir zwei neue Methoden zur Injektion von Wissen in mehrschichtige Sprachmodelle vor, um unterrepräsentiertes externes Wissen in Sprachrepräsentationen auszugleichen, sowie zwei neue Techniken zur Mitigation unfairer Verzerrungen.

Die nachfolgenden Kapitel der eingereichten Dissertation [La21] behandeln Fallstudien zur Adressierung jeder der einzelnen Herausforderungen.

2 Externes Wissen

Wie Moens [Mo18] diskutiert, ist unterrepräsentiertes Wissen (**C1**) eines der Hauptprobleme von heutigen Sprachrepräsentationen für CA.

Problembeschreibung. Distributionale Sprachrepräsentationen kodieren kein eindeutiges lexikalisch-semantisches Wissen und unterscheiden daher nicht zwischen semantischer *Relatedness* von Begriffen (z.B. *driver – car*) und echter *Similarity* (z.B. *car – vehicle*). Dieses Wissen ist jedoch manchmal bei Aufgaben zum argumentativen Verstehen, wie zum Beispiel beim argumentativen Schlussfolgern (Natural Language Inference) entscheidend. Das nachfolgende Prämisse-Hypothese-Paar [Wa19] verdeutlicht dies:

Prämisse	*Relation extraction systems populate knowledge bases with facts from unstructured text corpora.*
Hypothese	*Relation extraction systems populate knowledge bases with assertions from unstructured text corpora.*
Label	*Entailment*

Dieses Paar erfordert lexikalisches Wissen: Um die Inferenzaufgabe erfolgreich zu lösen, muss das Modell verstehen, dass die Begriffe *fact* und *assertion* im gegebenen Kontext als Synonyme dienen. Zwar gibt es linguistische Wissensbasen, aber sie bleiben ungenutzt.

Lösungsansatz. Wir schlagen zwei verschiedene Ansätze zur Injizierung von zwei verschiedenen Arten externen Wissens in kontextualisierte Embedding-Modelle vor: (1) Zunächst diskutieren wir, wie lexiko-semantisches Wissen durch ein zusätzliches Trainingsziel in der Pre-training-Phase in das Sprachmodell BERT [De19] kodiert werden kann. Unsere vorgeschlagene Erweiterung, „Lexically-Informed BERT (LIBERT)“ spezialisiert das Sprachmodell für wahre semantische Ähnlichkeit. Unsere Experimente zeigen, dass LIBERT eine bessere Performanz als das originale, lexikalisch-blinde, BERT bei mehreren Sprachverstehensaufgaben erzielt. Konkret übertrifft LIBERT BERT in 9 von 10 Aufgaben des sog. GLUE Benchmarks [Wa19], welcher u. A. grundlegende Aufgaben zum argumentativen Schlussfolgern enthält, und ist in der verbleibenden Aufgabe mit diesem gleichauf. Darüber hinaus zeigen wir konsistente Performanzgewinne auf drei Benchmarks zur lexikalischen Vereinfachung. (2) Zweitens schlagen wir einen Ansatz vor, um konzeptuelles Wissen und Wissen über Entitäten in der Welt *post hoc*, d.h. nach der Pre-training-Phase, in kontextualisierte Sprachmodelle zu injizieren. Konkret greifen wir hierbei auf Adapter-basiertes Training [Ho19] zurück. Hierbei werden neue, wesentlich kleinere Schichten für die Kodierung des zusätzlichen Wissens trainiert und im Gegensatz dazu werden die originalen Parameter des Modells nicht angepasst. Dieses Vorgehen hat verschiedene Vorteile: Zunächst ist das Training effizienter, da weniger Parameter optimiert werden. Als Ergebnis dessen wird entsprechend der Energieverbrauch im Training

verringert (**C5**). Des Weiteren wird das externe Wissen innerhalb der wenigen zusätzlichen Schichten gekapselt. Es kann daher zum einen je nach soziotechnischem Umfeld flexibel eingesetzt werden und zum anderen auch flexibel kombiniert werden. Wir demonstrieren die Effektivität beider Ansätze auf CA-Aufgaben und allgemeinen Textverstehensaufgaben.

3 Domänenspezifisches Wissen

Wie oben beschrieben existiert Argumentation in einer Vielfalt von Textdomänen. Im Zusammenspiel zwischen Sprachrepräsentationen und CA ist daher eine Herausforderung die Eignung der Repräsentationen für domänenspezifische Szenarios (**C2**).

Problembeschreibung. Je nach Textdomäne, auf die ein CA-Modell angewendet werden soll, können sich die beschriebenen Themen, die Genres und, damit zusammenhängend, der Grad der Formalität und das Vokabular, das in den Texten zu finden ist, unterscheiden. Beispielsweise werden auf einer Internetplattform, in der Restaurants diskutiert und bewertet werden, mehr Wörter verwendet werden, die im Zusammenhang mit *Essen* stehen, als in Finanzdokumenten, die sich mit Kryptowährung beschäftigen. Gleichzeitig können auch manche der Wörter, die zwischen zwei verschiedenen Domänen geteilt sind, andere Bedeutungen tragen: Im Beispiel der Finanzdokumente bezeichnet das Wort *„Bank“* mit hoher Wahrscheinlichkeit eine Finanzinstitution, während es sich in anderen Kontexten auch um eine Sitzgelegenheit handeln kann. Um die Eignung von Sprachrepräsentationen für solche Szenarien sicherzustellen, kann man sie auf spezifische Domänen anzupassen. Eine Möglichkeit dazu ist, zunächst domänenspezifische Daten zu identifizieren, basierend auf denen dann domänenspezifische Sprachrepräsentationen induziert werden können. Dies führt jedoch nicht immer zu einer verbesserten Performanz in Downstream-Anwendungen, da der Grad der Spezifizität einer bestimmten Domäne umgekehrt mit der verfügbaren Menge dieser Daten korreliert sein kann. Dies impliziert somit ein Abwägen von größeren und rauschbehafteteren vs. kleineren und homogeneren Datensätzen, welches letztendlich die Qualität der darauf induzierten Repräsentationen beeinflusst.

Lösungsansatz. Wir fokussieren uns auf die Spezifizität von Sprachrepräsentationen für den Bereich des wissenschaftlichen Schreibens. Die anspruchsvolle Natur [Gr17] und die einzigartigen Merkmale wissenschaftlicher Argumentation machen diese Domäne besonders interessant. Ein wichtiges argumentatives Werkzeug im wissenschaftlichen Schreiben ist das Referenzieren auf vorangegangene Arbeiten. Indem sie auf andere Arbeiten verweisen und anschließend von anderen zitiert werden, verknüpfen Forscher*innen ihre eigene Arbeit mit dem wissenschaftlichen Diskurs. Um die Herausforderung domänenspezifischen Wissens in numerischen Sprachrepräsentationen für CA besser zu verstehen, analysieren wir das Abwägen von größeren und rauschbehafteteren vs. kleineren und spezifischeren Korpora für die semantische Charakterisierung von Zitaten in Computerlinguistik-Publikationen.

Konkret studieren wir den Einfluss von generelleren versus generell wissenschaftlichen versus wissenschaftlichen Korpora aus der Computerlinguistik auf die daraus resultierenden Wortrepräsentationen. Zu diesem Zwecke fassen wir Polaritäts- und Funktionserkennung als Klassifikationsaufgaben und investigieren die Performanz von Convolutional Neural Networks mit allgemeineren und domänenspezifischeren Wortvektoren auf diesen Aufgaben. Unser bestes Modell übertrifft frühere Ergebnisse mit großen Abstand.

4 Geteiltes Wissen zwischen Aufgaben

Die Komplexität des CA-Feldes führt zu einer Vielfalt von meist isoliert betrachteten, tatsächlich aber miteinander verknüpften Problemen. Eine weitere Herausforderung ist daher die Frage nach dem Teilen von Wissen über verschiedene Aufgaben hinweg (**C3**).

Problembeschreibung. CA-Szenarien bieten sich oft natürlich für die gemeinsame Nutzung des Wissens an, welches in Sprachrepräsentationen kodiert ist. So ist beispielsweise das Identifizieren argumentativer Komponenten, wie z.B. von Prämissen und Behauptungen, mit dem Vorhersagen argumentativer Beziehungen, z.B. dass eine Prämisse eine bestimmte Behauptung unterstützt, verbunden. Eine klare gegenüber einer verworrenen argumentativen Struktur kann wiederum die wahrgenommene Qualität der Argumentation beeinflussen. In diesem Rahmen haben frühere Arbeiten die Effektivität von Multi-Task-Learning (MTL) bei Argumentationsaufgaben für ressourcenarme Szenarien gezeigt [Sc18].

Lösungsansatz. Wir verwenden solche induktiven Transferlern-Techniken, um zwei spezifische Probleme in CA zu lösen: (1) Zunächst untersuchen wir die Rolle feingranularer Argumentation in Bezug auf andere rhetorische Aspekte mit neuronalen MTL-Modellen. Zu diesem Zweck erweitern wir ein Korpus wissenschaftlicher Literatur um eine zusätzliche Argumentationsannotationsebene. Anschließend untersuchen wir zwei neuronale MTL-Architekturen, die auf gemeinsamen Recurrent Neural Networks basieren, und koppeln die neuronalen Architekturen mit einem gemeinsamen MTL-Ziel mit auf Unsicherheit basierender Gewichtung der aufgabenspezifischen Verluste [KGC18]. Wir validieren unseren Ansatz, indem wir zeigen, dass er traditionelle Modelle in Single-Task-Settings übertrifft. Schließlich zeigen wir, dass die Kopplung von rhetorischen Analyseaufgaben mit der feingranularen Extraktion argumentativer Komponenten unter Verwendung von MTL-Modellen die Ergebnisse für die rhetorischen Analyseaufgaben deutlich verbessert. (2) Dann bewegen wir uns vom Spezialfall der wissenschaftlichen Argumentation zu Argumentationsqualität in verschiedenen Bereichen des Online-Schreibens. Hier ist vor allem die theoriebasierte Perspektive [Wa17] im Gegensatz zur praktischen Perspektive noch untererforscht. In ersterer Perspektive wird die übergeordnete Argumentationsqualität als aus drei Unterdimensionen (Cogency, Effectiveness und Reasonableness) zusammengesetzt definiert, von denen jede wiederum aus einer Reihe von qualitätsbezogenen Aspekten besteht: (i) Cogency bezieht

sich auf die *logischen* Aspekte der Qualität von Argumenten. (ii) Effectiveness spiegelt die Überzeugungskraft eines Arguments wider und ist somit mit den *rhetorischen* Aspekten der argumentativen Qualität verbunden. (iii) Die Reasonableness gibt die Qualität eines Arguments im Kontext einer Debatte an und bezieht sich dabei auf die *dialektischen* Aspekte. Da es bislang kein groß angelegtes Korpus gibt, welches mit diesen theoriebasierten Qualitätsdimensionen annotiert ist, ist es auch nicht möglich, computergestützte Modelle zu trainieren, die die Komplementarität von Wissen über Aufgaben hinweg ausnutzen. Wir schließen diese Forschungslücke, indem wir das *GAQCorpus* vorstellen, das erste englische, theoriebasierte Korpus für Argumentationsqualität. Darüber hinaus demonstrieren wir Performanzverbesserungen in zwei Situationen, in denen die Komplementarität des Wissens in kontextualisierten Sprachrepräsentationen genutzt wird: (a) in einer flachen und einer hierarchischen MTL-Umgebung und (b) in einem sequenziellen Transfer.

5 Multilingualität

Basierend auf der Annahme, dass Argumentation in den meisten menschlichen Zivilisationen existiert, ist eine weitere Herausforderung Multilingualität (**C4**).

Problembeschreibung. Angesichts der großen Unterschiede bei der Verfügbarkeit der zum Training benötigten Ressourcen zwischen den Sprachen [Jo20] ist Multilingualität eine wesentliche Herausforderung für CA, da Systeme bei Eingabedaten aus bestimmten Sprachen schlecht oder gar nicht funktionieren, was wiederum bestimmte ethnische Gruppen systematisch ausschließt. Für Englisch – als ressourcenreiche Sprache – sind viele Datensätze verfügbar, die mit Labels für Aufgaben des argumentativen Verstehens annotiert sind und alle Bereiche von Aufgaben des argumentativen Verstehens abdecken. Für viele dieser Aufgaben und für viele Domänen gibt es jedoch keinen annotierten Datensatz in einer Vielzahl von Sprachen [To20]. Unannotierte Daten können für unüberwachte und selbstüberwachte Lernszenarien genutzt werden. Sie sind billiger zu beschaffen und liegen für viele Sprachen vor. Allerdings haben wir auch hier eine verzerrte Verteilung der Ressourcen: Vergleicht man die Größe der sprachspezifischen Wikipedias, die üblicherweise als vergleichbare Korpora für das Training mehrsprachiger Sprachrepräsentationen verwendet werden, so zählt Englisch als größte Wikipedia 6.184.229 Artikel, im Gegensatz zu Muscogee, einer der kleinsten, mit nur einem einzigen Artikel. Insgesamt wurden Artikel in nur 314 Sprachen erstellt.[3] Um das Problem zu lösen, untersuchen Forscher*innen effektive sprachübergreifende Transfertechniken [siehe RVS19], die im extremsten Fall, wenn keine Daten für die Zielaufgabe in der Zielsprache verwendet werden, als Zero-Shot-Transfer bezeichnet werden. In diesem Fall sind massiv mehrsprachige Transformer-Modelle, die durch Sprachmodellierung auf multilingualen Korpora vortrainiert wurden, z. B. multilingual BERT [De19], zu einem Standardparadigma in Natural Language Processing

[3] `https://en.wikipedia.org/wiki/List_of_Wikipedias` (4th of November, 2020)

geworden und bieten eine unübertroffene Transferleistung. Aktuelle Evaluierungen belegen jedoch ihre Effektivität nur für den Sprachtransfer (a) in Sprachen mit ausreichend großen Pretraining-Korpora und (b) zwischen linguistisch-nahen Sprachen.

Lösungsansatz. Wir analysieren die Grenzen des nachgelagerten Sprachtransfers mit massiv mehrsprachigen Transformer-Modellen und zeigen, dass sie, ähnlich wie statische sprachenübergreifende Wortrepräsentationen, in ressourcenarmen Szenarien und für linguistisch weit entfernte Sprachen wesentlich weniger effektiv sind. Unsere Experimente, die zwei semantisch anspruchsvolle Aufgaben mit argumentativem Schlussfolgern als Beispiel für eine CA-Aufgabe umfassen, korrelieren die Transferperformanz empirisch mit der linguistischen Ähnlichkeit zwischen Ausgangs- und Zielsprache, aber auch mit der Größe der beim Pre-training verwendeten Zielsprachenkorpora. Schließlich zeigen wir, dass ein kostengünstiger Few-Shot-Transfer (d.h. eine zusätzliche Feinabstimmung mit einigen wenigen zielsprachlichen Instanzen) in allen Bereichen effektiv ist.

6 Ethische Überlegungen

In früheren Forschungsarbeiten wurde auf mehrere ethische Probleme (**C5**) im Zusammenhang mit Sprachrepräsentationen hingewiesen.

Problembeschreibung. Der Schwerpunkt der zuvor skizzierten Herausforderungen liegt auf der Anpassung von Sprachrepräsentationen, um schließlich bessere „klassische“ Performanz in CA-Aufgaben zu erreichen. Wir erkennen jedoch an, dass unsere Systeme letztendlich in einem soziotechnischen Kontext eingesetzt werden, wodurch wir für mögliche Schäden im Zusammenhang mit der Art und Weise, wie wir Text numerisch darstellen, verantwortlich sind. Dies hat sich insbesondere bei CA-Anwendungen als kritisches Problem erwiesen [SW20]. Beispielhaft listen wir drei der wichtigsten ethischen Aspekte:

Exklusion. Frühere Arbeiten haben gezeigt, dass Natural Language Processing-Systeme oft nur die Bedürfnisse einer bestimmten Gruppe von Menschen erfassen. So gibt es für die meisten Sprachen der Welt keine existierenden Natural Language Processing-Modelle.

Ökologische Aspekte. Kürzlich zeigte Strubell et al. [SGM19], dass, bezogen auf den benötigten Energieaufwand, das Training vielschichtiger Sprachmodelle auf einer GPU einem transamerikanischen Flug gleichkommt. Diese Erkenntnis unterstreicht den hohen Energieverbrauch von state-of-the-art Natural Language Processing-Modellen.

Unfaire stereotypische Verzerrungen. Stereotypische Verzerrungen können aufgrund von Kookkurrenzverzerrungen in den Pre-training-Daten in Verbindung mit der distributionellen Natur von Sprachrepräsentationen entstehen: Kommt das Wort *„Frau“* häufiger in Verbindung mit dem Wort *„Familie“* vor als mit *„Karriere“* und das Wort *„Mann“* im Gegensatz

dazu häufiger mit dem Wort *„Karriere“* als mit *„Familie“*, kann das resultierende Modell und sein Output sexistisch verzerrt sein. Basierend auf der jeweiligen soziotechnischen Umgebung kann es dadurch zu ethischen Problemen kommen. Dies wurde als eine wesentliche Herausforderung für CA [SW20] hervorgehoben.

Lösungsansatz. In den vorangegangenen Kapiteln haben wir uns bereits indirekt mit den ersten beiden Aspekten befasst: (1) Um die Einbeziehung von Sprecher*innen anderer Sprachen als Englisch in CA-Technologien zu fördern, haben wir die inhärent mehrsprachige Natur von Argumentation anerkannt und das Ausmaß des Performanzverlustes analysiert, der beim derzeitigen state-of-the-art Paradigma des sprachübergreifenden Transfers entsteht. Anschließend haben wir einen ressourcenschonenden Ansatz zur Abschwächung dieser Verluste vorgeschlagen. Dieser Ansatz, die zielsprachliche Feinabstimmung in wenigen Schritten, berücksichtigt die (2) ökologischen Auswirkungen von Sprachtechnologien. Indem wir ressourcenschonende Methoden vorschlagen, können wir diesem Problem Rechnung tragen. Aus demselben Grund haben wir einen effizienten Ansatz für die Injektion externen Wissens vorgeschlagen. Darüber hinaus zielen wir darauf ab, den potenziellen Schaden, der durch CA-Technologien aufgrund unfairer stereotypischer Verzerrungen in Sprachrepräsentationen entsteht, zu mindern. Dazu stellen wir (1) zunächst XWEAT vor, eine Ressource, auf deren Grundlage wir eine multidimensionale Analyse von Verzerrungen in Sprachrepräsentationen durchführen. Unsere Analyse ist bis heute die umfangreichste Studie zu Verzerrungen in statischen Wortvektorräumen. Im Rahmen dieser zeigen wir beispielsweise, dass das zu erwartende Ausmaß einer Verzerrung in einem bilingualen Vektorraum ungefähr dem Mittel der entsprechenden monolingualen Räume entspricht. Anschließend (2) stellen wir ein allgemeines Framework vor, welches frühere Arbeiten zur Evaluation und Mitigation von Verzerrungen in statischen Wortrepräsentationen zusammenfasst. Innerhalb dieses Frameworks schlagen wir ein neues Verzerrungsmaß (BAT) und drei Methoden zur Mitigation solcher stereotypischer Verzerrungen in Wortvektorräumen (GBDD, BAM und Explicit Neural Debiasing (DebiasNet)) vor.

7 Fazit

Die Komplexität menschlicher Argumentation verlangt nach fortschrittlichsten Sprachtechnologien. Hierbei wurden Sprachrepräsentationen als ein Hauptengpass identifiziert. Jedoch existiert bis heute keine systematische Forschung am Zwischenspiel von Repräsentationslernen und CA. In der vorliegenden Dissertation adressierten wir diese Forschungslücke. Zu diesem Zwecke wurden fünf Herausforderungen auf Basis inhärenter Charakteristika von Argumentation identifiziert und in Fallstudien untersucht. Im Rahmen dessen wurden neue Ressourcen und Analysen präsentiert sowie Maße und Methoden vorgeschlagen. Als wissenschaftliche Gemeinschaft, die CA-Technologien entwickelt, sind wir dafür verantwortlich, effektive, faire, inklusive und nachhaltige Verfahren zu gewährleisten. Wir hoffen, dass die vorgestellten Ergebnisse weitere Forschung zur Erreichung dieses Ziels inspirieren.

Literatur

[Ar06] Aristotle: On Rhetoric: A Theory of Civic Discourse. Translated by George A. Kennedy, Oxford University Press, Oxford, UK, ca. 350 B.C.E./ translated 2006, ISBN: 978-0195305098.

[AR11] Abu-Jbara, A.; Radev, D.: Coherent Citation-Based Summarization of Scientific Papers. In: Proceedings of ACL. ACL, Portland, Oregon, USA, S. 500–509, 2011.

[De19] Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: Proceedings of NAACL-HLT. ACL, Minneapolis, Minnesota, S. 4171–4186, 2019.

[Gr17] Green, N.: Manual Identification of Arguments with Implicit Conclusions Using Semantic Rules for Argument Mining. In: Proceedings of ArgMining. ACL, Copenhagen, Denmark, S. 73–78, 2017.

[Gr20] Gretz, S.; Friedman, R.; Cohen-Karlik, E.; Toledo, A.; Lahav, D.; Aharonov, R.; Slonim, N.: A Large-Scale Dataset for Argument Quality Ranking: Construction and Analysis. In: The Thirty-Fourth AAAI. AAAI Press, New York, NY, USA, S. 7805–7813, 2020.

[HG16] Habernal, I.; Gurevych, I.: Which argument is more convincing? Analyzing and predicting convincingness of Web arguments using bidirectional LSTM. In: Proceedings of ACL. ACL, Berlin, Germany, S. 1589–1599, 2016.

[Ho19] Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; de Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; Gelly, S.: Parameter-Efficient Transfer Learning for NLP. In (Chaudhuri, K.; Salakhutdinov, R., Hrsg.): Proceedings of ICML. Bd. 97, PMLR, Long Beach, CA, USA, S. 2790–2799, 2019.

[Jo20] Joshi, P.; Santy, S.; Budhiraja, A.; Bali, K.; Choudhury, M.: The State and Fate of Linguistic Diversity and Inclusion in the NLP World. In: Proceedings of ACL. ACL, Online, S. 6282–6293, 2020.

[KGC18] Kendall, A.; Gal, Y.; Cipolla, R.: Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In: CVPR 2018. IEEE Computer Society, Salt Lake City, UT, USA, S. 7482–7491, Juni 2018.

[La21] Lauscher, A.: Language Representations for Computational Argumentation, Diss., Mannheim, Germany: University of Mannheim, 2021.

[Mi13] Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; Dean, J.: Distributed Representations of Words and Phrases and their Compositionality. In (Burges, C. J. C.; Bottou, L.; Ghahramani, Z.; Weinberger, K. Q., Hrsg.): NeurIPS. Lake Tahoe, Nevada, USA, S. 3111–3119, 2013.

[Mo18] Moens, M.-F.: Argumentation mining: How can a machine acquire common sense and world knowledge? Argument & Computation 9/1, S. 1–14, 2018, ISSN: 1946-2174, 1946-2166.

[RVS19] Ruder, S.; Vulić, I.; Søgaard, A.: A Survey of Cross-lingual Word Embedding Models. Journal of Artificial Intelligence Research 65/, S. 569–631, 2019, ISSN: 1076-9757.

[Sc18] Schulz, C.; Eger, S.; Daxenberger, J.; Kahse, T.; Gurevych, I.: Multi-Task Learning for Argumentation Mining in Low-Resource Settings. In: Proceedings of NAACL-HLT. ACL, New Orleans, Louisiana, S. 35–41, 2018.

[SGM19] Strubell, E.; Ganesh, A.; McCallum, A.: Energy and Policy Considerations for Deep Learning in NLP. In: Proceedings of ACL. ACL, Florence, Italy, S. 3645–3650, 2019.

[SW20] Spliethöver, M.; Wachsmuth, H.: Argument from Old Man's View: Assessing Social Bias in Argumentation. In: Proceedings of ArgMining. ACL, Online, S. 76–87, 2020.

[To20] Toledo-Ronen, O.; Orbach, M.; Bilu, Y.; Spector, A.; Slonim, N.: Multilingual Argument Mining: Datasets and Analysis. In: Findings of EMNLP. ACL, Online, S. 303–317, 2020.

[Wa17] Wachsmuth, H.; Naderi, N.; Hou, Y.; Bilu, Y.; Prabhakaran, V.; Thijm, T. A.; Hirst, G.; Stein, B.: Computational Argumentation Quality Assessment in Natural Language. In: Proceedings of EACL. ACL, Valencia, Spain, S. 176–187, 2017.

[Wa19] Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S. R.: GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In: ICLR. OpenReview.net, New Orleans, LA, USA, Mai 2019.

Anne Lauscher wurde am 11. Juli 1991 in Trier geboren. Sie erlangte den B.Sc. in Wirtschaftsinformatik an der DHBW Mannheim (*ETCS Ranking: A*) in Kooperation mit SAP und den M.Sc. in Wirtschaftsinformatik an der Universität Mannheim (*mit Auszeichnung*). Im Rahmen ihrer Promotion in Natural Language Processing (NLP) in der Data and Web Science Group der Universität Mannheim (*summa cum laude*) erforschte sie numerische Sprachrepräsentationen. Währenddessen arbeitete sie zudem als Forschungspraktikantin und unabhängige Forscherin für Grammarly und das Allen Institute for Artificial Intelligence. Aktuell ist sie Postdoktorandin in der Data and Marketing Insights Unit der Bocconi Universität in Mailand, Italien. Hier fokussiert sie sich auf die Integration demographischer Faktoren in Dialogsysteme, um Systemperformanz und Fairness zu erhöhen. Ihre Forschung wurde auf internationalen top-tier Konferenzen für NLP und KI publiziert und ausgezeichnet. Zuletzt erhielt sie den Maria Gräfin von Linden-Preis, der exzellente Nachwuchswissenschaftlerinnen in Baden-Württemberg würdigt.

Sicherheitsaspekte von Optimierungen in Mikroarchitekturen bei Software-Angriffen[1]

Moritz Lipp[2]

Abstract: Um der Komplexität moderner Computersysteme gerecht werden zu können, werden mithilfe von Abstraktion die Implementierungsdetails von Software- und Hardwarekomponenten versteckt. Die dadurch ermöglichten unterschiedlichen Implementierungen können sich in Aspekten wie Leistung, Sicherheit, Energieeffizienz oder anderen Eigenschaften unterscheiden. Mikroarchitekturangriffe nutzen kleine Variationen dieser Eigenschaften in moderner Prozessoren aus, um sensible Informationen, die auf dem System verarbeitet werden, zu stehlen.

In dieser Arbeit erweitern wir diese softwarebasierte Mikroarchitekturangriffe und Gegenmaßnahmen: Wir identifizieren bisher unbekannte Angriffsvektoren, die die grundlegendsten Sicherheitsgarantien moderner Prozessoren umgehen und weltweit zu Veränderung in Betriebssystemen und Prozessoren führen. Wir kombinieren traditionelle physikalische Seitenkanalanalysen mit softwarebasierten Mikroarchitektur-Angriffstechniken, um geheime kryptografische Schlüssel zu entwenden und geben neue Einblicke in die Entwicklung effektiver und dauerhaft nachhaltige Gegenmaßnahmen, die in allen modernen Betriebssystemen umgesetzt wurden.

1 Einleitung

In der Softwareentwicklung und Informatik bezeichnen Schnittstellen definierte Grenzen, welche zwischen zwei oder mehreren Komponenten geteilt sind und es diesen ermöglichen miteinander zu interagieren. Diese Komponenten existieren in Software und Hardware oder können eine Kombination aus beidem sein. Durch die genaue Spezifikation wie Komponenten miteinander interagieren sollen und welche Erwartungen man einer Komponente gegenüber haben kann, ermöglichen Schnittstellen eine hohe Flexibilität in der Implementierung der Komponenten so lang sich diese so verhalten wie festgelegt.

Durch die Abstraktion von Komponenten und die Verwendung von einfachen Schnittstellen wird nicht nur die Komplexität des Systems verschleiert, sondern auch die Implementierung von unterschiedlichsten Varianten ermöglicht. Für ein Software-Projekt wird beispielsweise ein neuer, effizienter Algorithmus für eine definierte Schnittstelle implementiert. Durch die Einhaltung der Spezifikation kann die alte Implementierung, ohne andere Stellen des Systems zu ändern, einfach ausgetauscht werden. Diese Schnittstellen existieren nicht nur in Softwareprojekten, sondern auch zwischen einzelnen Programmen, dem Betriebssystem

[1] Englischer Titel der Dissertation: “Exploiting Microarchitectural Optimizations from Software”

[2] Technische Universität Graz, IAIK, Inffeldgasse 16a, 8010 Graz, Österreich, mail@mlq.me

und der eigentlichen Hardware. Obwohl die einzelnen Implementierungen alle den gleichen Zweck erfüllen, unterschieden sie sich in anderen Aspekten, wie beispielsweise der Performance.

Architektur vs. Mikroarchitektur.

In Computing bezeichnet eine Architektur typischerweise die 'Instruction Set Architecture' (ISA) oder Computerarchitektur. Während die ISA als Schnittstelle zwischen dem Prozessor und der Software, der auf diesem läuft, dient, bezeichnet die Computer-Mikroarchitektur die eigentliche Hardwareimplementierung der Architektur [HP17]. Dadurch wird die Komplexität der Implementierung durch die (weniger komplexe) Spezifikation der Architektur versteckt. Verschiedene Mikroarchitekturen implementieren die gleiche Spezifikation der Architektur auf unterschiedliche Art und Weise: Dadurch kann es passieren, dass dieselbe Applikation sich unterschiedlich verhält. Obwohl die berechneten Resultate dieselben sind, können sich die Ausführungszeit, der Stromverbrauch oder andere physikalische Eigenschaften während der Ausführung unterscheiden.

Allerdings ist diese Ansicht nicht auf den Prozessor alleine limitiert, sondern trifft auf jede Abstraktionsebene zu. Das beinhaltet auch andere Hardwareschnittstellen; wie zum Beispiel den Arbeitsspeicher. Solange der Speicher sich an die Spezifikation hält, kann der Hersteller die Implementierung auf unterschiedliche Art und Weisen durchführen. Des Weiteren ist diese Betrachtungsweise nicht auf Hardware alleine eingeschränkt. Zum Beispiel definiert ein Betriebssystem Schnittstellen für Anwendungsprogramme, damit diese mit dem Betriebssystem kommunizieren können. Das bedeutet, dass unterschiedliche Betriebssysteme als unterschiedliche Mikroarchitekturen betrachtet werden können, da sich die Implementierung von Betriebssystemen unterscheiden kann und dennoch die funktionale Korrektheit gegeben bleibt.

Optimierungen hinter verschlossenen Türen.

In der Vergangenheit wurden moderne Prozessoren einzig und allein hinsichtlich Leistung und Stromverbrauch optimiert. Da Endkunden immer schneller und schneller werdende Prozessoren verlangen, müssen die Hersteller die Prozessoren optimieren. In einer Welt, die durch Benchmark-Tests geprägt ist, zählt jeder Taktzyklus und spielt eine maßgebliche Rolle am Marktanteilen der Hersteller. Darüber hinaus versucht die Software selbst das meiste aus der Plattform herauszuholen.

Eine Möglichkeit, die Leistung zu steigern, ist für den am häufigsten auftretenden Fall, zu optimieren. Wenn dieser Fall effizienter abgearbeitet und somit schneller ausgeführt werden kann als die weniger häufigen Fälle (wie Spezialfälle oder Fehler), dann wird die Leistung gesteigert.

Die klaren Grenzen, die durch Schnittstellen definiert werden, ermöglichen es *alles* in der darunterliegenden Implementierung zu machen solange das Verhalten der Beschreibung des Interfaces entspricht. Das eröffnet den Entwicklern unzählige Möglichkeiten das Interface zu implementieren und somit gleichzeitig es für verschiedene Faktoren und in allen erdenklichen Weisen zu optimieren.

Es gibt verschiedene Möglichkeiten die Leistung einer Applikation, die auf einem System ausgeführt wird, zu erhöhen: Mit *Compiler Optimierungen* wird der Quellcode transformiert und anhand von verschiedenen Attributen optimiert [Mo98]. Mit sogenannten *Laufzeit-Optimierungen* durch verschiedene Interpreter wird das Programm während der Laufzeit abhängig von der aktuellen Arbeit optimiert. Das *Betriebssystem*, auf dem das Programm ausgeführt wird, kann dem Programm nicht nur die Ressourcen erhöhen aber auch die Art und Weise wie mit diesen Ressourcen umgegangen wird verbessern. Eine wichtige Aufgabe des Betriebssystems ist es den Speicher des Systems zu verwalten. Um dies umzusetzen, werden notwendige Strukturen für Prozesse angelegt denen einzelne Segmente des Speichers zugewiesen werden. Es gibt verschiedenste Optimierungstechniken, die es erlauben den Speicherverbrauch von Programmen zu optimieren und die allgemeine Leistung zu erhöhen. Diese Techniken sind alle transparent für die ausgeführten Programmen. Des Weiteren kann die eigentliche *Hardware*, also der Prozessor der das Programm ausführt, optimiert werden. Neben der Taktrate des Prozessors, gibt es noch viele andere Faktoren, die die Leistung beeinflussen: Verschiedene Größen und Eigenschaften von sogenannten Caches spielen eine genauso wichtige Rolle wie die Art und Weise wie der Prozessor Instruktionen abarbeitet [SL13]. Des Weiteren beeinflusst der Einsatz von verschiedenen Vorhersagemechanismen die Leistung des Prozessors drastisch [HP17].

Mit der Absicht, die gesamte Leistung des Systems zu verbessern, wurden auf der anderen Seite verschiedene Sicherheitsgarantien der Architektur in der Implementierung vernachlässigt. Da das Innenleben der Mikroarchitektur hinter der Schnittstelle versteckt ist und von außen nicht inspiziert werden kann, verhält sich das System wie es erwartet wird, solange die Annahmen auf der architekturellen Ebene eingehalten werden. In dieser Arbeit fokussieren wir uns auf die Auswirkungen dieser Optimierungen auf die Sicherheit des Systems.

1.1 Durch das Opake blicken.

Das Definieren klarer Schnittstellen und das Verstecken der Komplexität hat zur Annahme geführt, dass Implementierungen wie eine undurchsichtige Blackbox erscheinen, und deren innere Details nicht inspizieren werden können. Diese Annahme erlaubt es sogar die Sicherheitsgarantien in der Mikroarchitektur völlig zu ignorieren, um die Leistung zu verbessern [SHW11].

Allerdings kann Information nicht nur über *legitime Kanäle*, die durch die Schnittstelle definiert wurden, übertragen werden, sondern eine Implementierung kann zusätzliche Information über sogenannte *nebensächliche Kanäle* Preis geben. Dazu zählen beispielsweise

die Antwortzeit oder der Stromverbrauch der Implementierung. Diese Kanäle dienen als *Seitenkanäle*, wenn das System Informationen über diesen Kanal verrät. Sogenannte Seitenkanalangriffe erlauben es Angreifern diese Information auszunutzen.

Mit Seitenkanalangriffen und Fault-Angriffen gibt es nicht-invasive und invasive Möglichkeiten von einer Implementierung zu lernen oder mit dieser so zu hantieren, dass sensitive Daten abgegriffen werden können [MOP08]. Obwohl diese Angriffe typischerweise physikalischen Zugriff auf die Hardware benötigen, konzentrieren wir uns in dieser Arbeit auf Techniken, die es ermöglichen solche Angriffe rein von Software auszuführen.

Seitenkanalangriffe. Durch die Annahme, dass das Innenleben einer Mikroarchitektur von außen nicht inspiziert werden kann, werden Seitenkanalangriffe meistens im Angreifermodell für Prozessoren außer Acht gelassen. Dennoch erlauben Seitenkanalangriffe Informationen, die die Implementierung eines Systems in Software oder Hardware Preis gibt, auszunutzen. Typischerweise werden dabei physikalische Eigenschaften, wie der Stromverbrauch oder die magnetische Abstrahlung beobachtet und die erhaltenen Messungen dafür verwendet um auf andere, anderwärtig nicht zugreifbare, sensitive Informationen zu schließen. Zum Beispiel: Um eine kryptografische Berechnung auszuführen, muss der Prozessor in manchen Algorithmen eine Operation ausführen, die mehr Strom verbraucht, wenn ein gesetztes Bit im geheimen Schlüssel verarbeitet wird als wenn ein nicht-gesetztes Bit verarbeitet wird. Anhand des Stromverbrauchs kann der Angreifer die Strommessung mit der verwendeten Operation korrelieren und somit den gesamten geheimen Schlüssel herausfinden [MOP08]. Der Angreifer beobachtet nur indirekte Information (den Stromverbrauch) über den geheimen Schlüssel, kann diesen aber nicht direkt auslesen.

In der Vergangenheit benötigten Seitenkanalangriffe physikalischen Zugriff auf das angegriffene Gerät um Sonden oder andere Peripherien anzubringen, die die Messungen durchführten [MOP08]. In den letzten Jahren hingegen sind allerdings einige softwarebasierte Angriffe veröffentlicht worden, wodurch die Notwendigkeit, physikalischen Zugriff auf das Gerät zu haben gelockert wurde [Sp17]. Viele dieser Mikroarchitektur-Angriffstechniken zielen auf den Cache des Prozessors ab. Diese kleinen aber schnellen Speicher optimieren unsichtbar den Zugriff auf oft benutzten Speicher. Die Zeit-Unterschiede der Zugriffszeiten auf Speicher, der in einem Cache gepuffert wird oder erst nachgeladen werden muss, ermöglicht es Angreifern Seitenkanalangriffe in Software zu implementieren. Diese sogenannten Cache-Angriffe ermöglichen es kryptografische Algorithmen zu brechen [YF14], den Benutzer zu beobachten [Li16] und virtuelle Maschinen auszuspionieren.

Fault-Angriffe. Mit Fault-Angriffen bringt ein Angreifer ein Gerät für sehr kurze Zeit außerhalb des physikalischen Bereichs für die das Gerät spezifiziert ist. Durch die kurzzeitige Änderung der Stromspannung, sehr hohe oder sehr niedrige Temperaturen, Strahlung oder durch den Beschuss mit Lasern können Fehler in Berechnungen induziert werden. Wenn es allein durch Software möglich ist, ein Gerät außerhalb des vorgesehenen Bereichs zu

bringen, sind softwarebasierte Fault-Angriffe möglich. Ein Beispiel ist der sogenannte Rowhammer-Fehler [Ki14]: Durch viele Zugriffe auf spezifische Speicherstellen wird der Arbeitsspeicher außerhalb der vorgesehenen Bedingungen verwendet und es können sich Daten in benachbarten Speicherzellen ändern. Diese sogenannten *Bitflips* konnten von Angreifern so ausgenutzt werden, um Administrationsrechte auf dem Gerät zu erlangen.

2 Wissenschaftliche Beiträge

Die wissenschaftlichen Beiträge dieser Arbeit bringen den Stand der Technik von Mikroarchitektur-Angriffstechniken und Gegenmaßnahmen maßgeblich voran:

- **Entdeckung von *Transient-Execution Angriffen*.** Während wir mit dem Meltdown-Angriff [Li18] die transiente Ausführung von Instruktionen, bevor ein Fehler in einer *out-of-order* CPU behandelt wird, ausnützen, nützen wir mit Spectre [Ko19] die transiente Ausführung von Instruktionen nach einer falschen Vorhersage aus. Dieser Angriff erlaubt es uns die fundamentalsten Sicherheitsgarantien von modernen Prozessoren zu umgehen und im Gegensatz zu Seitenkanalangriffen direkt sensitive Daten auszulesen. Durch unsere Entdeckung von Meltdown und Spectre ist ein eigenes Feld innerhalb der Mikroarchitekturangriffe entstanden, so genannte Transient-Execution Angriffe.
- **Identifizierung von *bislang unbekannten Angriffsvektoren*.** Mit Takeaway [Li20a] haben wir die Funktion des Cache-Wege-Prediktors von AMD Prozessoren rekonstruiert und dadurch zwei neue Angriffstechniken entdeckt, die es ermöglichen sensitive Informationen zu stehlen.
- **Untersuchung ob bestehende Angriffstechniken *remote* ausgenützt werden können.** Mit Nethammer [Li20b] haben wir gezeigt, dass es möglich ist, Rowhammer Fehler auf Computern einzig allein über das Netzwerk zu induzieren ohne das der Angreifer Schadcode auf dem Zielgerät ausführen muss. Des Weiteren zeigen wir, dass ein Interrupt-basierter Angriff auch in JavaScript in einem Webbrowser ausgeführt werden kann, der es dem Angreifer ermöglicht die Zeitpunkte herauszufinden, wenn der Benutzer des Geräts eine Taste drückt. Diese genauen Zeitpunkte erlauben es dem Angreifer dann, Passwörter oder PINs, sowie besuchte Webseiten, herauszufinden.
- **Kombination von *traditioneller physikalischer Seitenkanalanalyse* und *modernen softwarebasierten Mikroarchitektur-Angriffstechniken***. Mit PLATYPUS [Li21b] beobachten wir den Stromverbrauch eines Prozessors nur durch Software, um sensitive Daten wie kryptografische Schlüssel zu entwenden. Obwohl die Abtastrate der Schnittstelle viel geringer ist als die von Oszilloskopen, nutzen wir Techniken von Mikroarchitekturangriffen aus um diese Limitierungen zu überwinden.
- **Gewinn von *neuen Erkenntnissen* für effiziente Gegenmaßnahmen.** Es ist sehr wichtig, neue Angriffsvektoren zu entdecken und im Detail zu studieren, um ein

besseres Verständnis über die Anforderungen an effiziente Gegenmaßnahmen zu gewinnen Mit KAISER [Gr17] haben wir eine verstärkte Isolierung für Prozesse vorgeschlagen, um vor existierenden Seitenkanalangriffen zu schützen. Nachträglich stellte sich heraus, dass KAISER auch eine Gegenmaßnahme gegen den Meltdown-Angriff darstellt und keine Änderungen an der existierenden Hardware benötigt. Somit wurde die Idee von KAISER adaptiert und in jedem bekannten Betriebssystem implementiert.

Während der folgende Inhalt die individuellen Beiträge dieser Arbeit beschreibt, bezieht er sich dennoch nur auf eine Untermenge meiner wissenschaftlichen Beiträge, die ich während meines Doktorats geleistet habe. Eine ausführliche Beschreibung dieser Ergebnisse findet sich in der verfassten Doktorarbeit [Li21a]. Die folgenden wissenschaftlichen Publikationen, die in dieser Doktorarbeit inkludiert sind, treiben den Stand der Technik von Mikroarchitektur-Angriffstechniken und Gegenmaßnahmen maßgeblich voran:

Take a Way: Exploring the Security Implications of AMD's Cache Way Predictors. Um den Stromverbrauch zu reduzieren und die Leistung der Prozessoren zu erhöhen nutzt AMD einen Wege-Prediktor für den ersten Level des Datencaches, der vorhersagt in welchem Weg sich eine bestimmte Adresse befindet. Die Implementierung markiert jede Cache-Zeile mit einem sogenannten Micro-Tag, der mittels einer nicht dokumentierten Hashfunktion berechnet wird. In dieser Publikation [Li20a] rekonstruieren wir diese Hashfunktion in allen Mikroarchitekturen von 2011 bis 2019 und präsentieren zwei neue Angriffstechniken, die den Wege-Prediktor als Seitenkanal ausnützen. Mittels dieser Techniken demonstrieren wir einen versteckten Kommunikationskanal, einen Angriff auf Randomisierungstechniken, rekonstruieren kryptografische Schlüssel und kombinieren den Seitenkanal mit einem Spectre-Angriff, um sensitive Daten vom Betriebssystem zu lesen.

Meltdown: Reading Kernel Memory from User Space. Durch die Annahme, dass der mikroarchitekturelle Zustand unsichtbar ist und das mikroarchitekturelle Zustandsänderungen nicht beobachtet werden können, können Sicherheitsgarantien während der transienten Ausführung von Instruktionen vernachlässigt werden, da diese keine sichtbaren Konsequenzen haben. Da mikroarchitekturelle Seitenkanalangriffe nur Metadaten über die Ausführung eines Programmes erkennen können, spielen diese keine bedeutende Rolle im Angreifermodell für Prozessoren. In dieser Publikation [Li18] präsentieren wir den Meltdown-Angriff, der die Vernachlässigung der Berechtigungsüberprüfung während der transienten Ausführung von Instruktionen ausnützt. Dieser Angriff ermöglicht es Daten, auf die architekturell nicht zugegriffen werden kann, mit Hilfe eines Seitenkanals sichtbar zu machen.

Zusammen mit dem Spectre-Angriff [Ko19] beschreibt Meltdown eine neue Klasse von Mikroarchitekturangriffen und legt den Weg für eine Reihe an weiteren Angriffen dieser

Art und der Forschung von notwendigen Gegenmaßnahmen. Mit Foreshadow [Va18], Zombieload [Sc19a], RIDL [Sc19b], Fallout [Ca19], CrossTalk [Ra21], LazyFP [SP18], CacheOut [Sc20] und Medusa [Mo20] folgten viele weitere Angriffe, die die Wichtigkeit und das immense weltweite Interesse an diesen Angriffen unterstreicht.

Practical Keystroke Timing Attacks in Sandboxed JavaScript. Um die genauen Zeitpunkte von Tastatur- oder Touchpadeingaben beobachten zu können, musste ein Angreifer bisher entweder physikalischen Zugriff auf das Gerät besitzen oder selbst Code auf dem Gerät ausführen können. In dieser Publikation [Li17] haben wir untersucht, ob ein anderer unserer Angriffe [Sc18] auch mit Hilfe von JavaScript innerhalb einer Webseite ausgeführt werden kann. Wir zeigen, dass man nicht nur Tastatureingaben auf Desktop-Maschinen erkennen kann, sondern auch Berührungen auf einem Mobilgerät, wie beispielsweise die PIN- oder Passworteingabe.

KASLR is Dead: Long Live KASLR. Intel empfiehlt aus Performance-Gründen, dass das Betriebssystem in den Adressraum jedes Prozesses vollständig integriert werden soll [In19]. Ein Bit in den Paging-Strukturen definiert zur Sicherheit, dass auf Betriebssystemspeicher durch ein Programm nicht zugegriffen werden darf. Dadurch wird der Speicherbereich virtuell geteilt: Den Speicher für den Betriebssystem-Modus und den Speicher für den Programm-Modus.

Dieser alleinige Adressraum ermöglichte Seitenkanalangriffe, die einen Sicherheitsmechanismus von Betriebssystemen aushebeln konnten [Gr16; HWH13; JLK16]. Mit KAISER [Gr16] präsentierten wir eine Änderung des Betriebssystems, das 2 Adressräume verwendete: Einen für das Betriebssystem und einen für das Programm. Wir implementierten die Idee für das Linux Betriebssystem und evaluierten, dass damit diese Seitenkanalangriffe erfolgreich verhindert werden können. Inzwischen zeigte sich aber der Ansatz von der noch-strikteren Trennung nützlich, da diese Implementierung auch gegen den Meltdown-Angriff schützt. Somit wurde unsere Idee in allen bekannten Betriebssystemen umgesetzt.

Nethammer: Inducing Rowhammer Faults through Network Requests. In der Vergangenheit wurde der Rowhammer-Angriff immer als ein lokaler Angriff beschrieben, bei dem der Angreifer die Möglichkeit benötigt, auf dem Computer Code auszuführen oder das Opfer auf eine Webseite lockt, wo Schadcode in JavaScript Bitflips indiziert. In dieser Publikation zeigen wir, dass es möglich ist, Bitflips auf einem Computer über das Netzwerk auszulösen ohne dass der Angreifer selbst auch nur eine einzige Zeile Code ausführen muss. Durch spezielle, sehr kleine Netzwerkpakete ist es uns möglich, Bitflips über das Netzwerk zu induzieren. Des Weiteren zeigen wir in der Arbeit, dass bestimmte Gegenmaßnahmen in der Hardware nicht ausreichend vor diesen Angriffen schützen.

PLATYPUS: Software-based Power Side-Channel Attacks on x86. Bei einem klassischen Stromseitenkanalangriff benötigt ein Angreifer physikalischen Zugriff auf das Zielgerät um den Stromverbrauch mittels eines Oszilloskops zu messen. Damit der Prozessor innerhalb der Strombedingungen läuft, bieten Prozessoren eine Schnittstelle an, um den aktuellen Stromverbrauch vom Prozessor und dem Arbeitsspeicher zu messen. Auf Intel Prozessoren ermöglicht das *Intel Running Average Power Limit* (RAPL), diese Messungen in Software auszulesen. Obwohl die Auflösung dieser Schnittstelle im Vergleich zu einem Oszilloskop sehr gering ist, demonstrieren wir, dass man mit statistischer Evaluierung Veränderungen im Stromverbrauch beobachten kann.

Mit dem PLATYPUS Angriff [Li21b] nützen wir den uneingeschränkten Zugriff auf diese Schnittstelle mit verschiedenen Techniken aus, um erfolgreich kryptografische Schlüssel aus SGX Enclaven und dem Betriebssystem zu entwenden.

3 Schlussfolgerung

In dieser Arbeit zeigen wir das Optimierungen in der Mikroarchitektur mittels Software auf allen Abstraktionsebenen ausgenützt werden können. Zudem legen wir dar, dass Seitenkanalangriffe und Fault-Angriffe das Entwenden von sensitiven Informationen über alle Abstraktionsebenen ermöglichen. Mit Angriffen, die einzig und allein in Software implementiert werden, decken wir geheime Informationen auch remote auf.

Durch unsere Arbeit an Meltdown und Spectre entstand eine komplett neue Klasse an Mikroarchitekturangriffen, durch die viele weitere internationale Arbeiten zu dieser Thematik in der wissenschaftlichen Community hervorgebracht wurden. Dadurch, dass diese Angriffe im Gegensatz zu Seitenkanalangriffen direkt Daten abgreifen können, sind Gegenmaßnahmen auf verschiedenen Ebenen notwendig: Die Kombination von notwendigen Updates von Prozessoren, Betriebssystemen und Compilern gehen meistens mit nicht zu vernachlässigen Leistungseinbußen einher. Mit hoher Wahrscheinlichkeit existieren ähnliche Sicherheitslücken aber in allen ausreichend komplexen Systemen. Aus diesem Grund wird die Zukunft nicht nur weitere Angriffstechniken zu Tage bringen, aber auch ein Umdenken erzwingen wie man diesen Angriffen auf der Mikroarchitekturebene entgegenwirken muss. Künftige Optimierungen können und werden in neuen Mikroarchitekturen implementiert und zukünftige Forschung wird sich mit den Sicherheitsimplikationen dieser beschäftigen.

Literatur

[Ca19] Canella, C.; Genkin, D.; Giner, L.; Gruss, D.; Lipp, M.; Minkin, M.; Moghimi, D.; Piessens, F.; Schwarz, M.; Sunar, B.; Van Bulck, J.; Yarom, Y.: Fallout: Leaking Data on Meltdown-resistant CPUs. In: ACM CCS. 2019.

[Gr16] Gruss, D.; Maurice, C.; Fogh, A.; Lipp, M.; Mangard, S.: Prefetch Side-Channel Attacks: Bypassing SMAP and Kernel ASLR. In: ACM CCS. 2016.

[Gr17] Gruss, D.; Lipp, M.; Schwarz, M.; Fellner, R.; Maurice, C.; Mangard, S.: KASLR is Dead: Long Live KASLR. In: ESSoS. 2017.

[HP17] Hennessy, J. L.; Patterson, D. A.: Computer Architecture: A Quantitative Approach. Morgan Kaufmann, 2017.

[HWH13] Hund, R.; Willems, C.; Holz, T.: Practical Timing Side Channel Attacks against Kernel Space ASLR. In: IEEE S&P. 2013.

[In19] Intel: Intel 64 and IA-32 Architectures Software Developer's Manual, Volume 3 (3A, 3B & 3C): System Programming Guide, 2019.

[JLK16] Jang, Y.; Lee, S.; Kim, T.: Breaking Kernel Address Space Layout Randomization with Intel TSX. In: ACM CCS. 2016.

[Ki14] Kim, Y.; Daly, R.; Kim, J.; Fallin, C.; Lee, J. H.; Lee, D.; Wilkerson, C.; Lai, K.; Mutlu, O.: Flipping Bits in Memory Without Accessing Them: An Experimental Study of DRAM Disturbance Errors. In: ISCA. 2014.

[Ko19] Kocher, P.; Horn, J.; Fogh, A.; Genkin, D.; Gruss, D.; Haas, W.; Hamburg, M.; Lipp, M.; Mangard, S.; Prescher, T.; Schwarz, M.; Yarom, Y.: Spectre Attacks: Exploiting Speculative Execution. In: IEEE S&P. 2019.

[Li16] Lipp, M.; Gruss, D.; Spreitzer, R.; Maurice, C.; Mangard, S.: ARMageddon: Cache Attacks on Mobile Devices. In: USENIX Security Symposium. 2016.

[Li17] Lipp, M.; Gruss, D.; Schwarz, M.; Bidner, D.; Maurice, C.-m.-t.-n.; Mangard, S.: Practical Keystroke Timing Attacks in Sandboxed JavaScript. In: ESORICS. 2017.

[Li18] Lipp, M.; Schwarz, M.; Gruss, D.; Prescher, T.; Haas, W.; Fogh, A.; Horn, J.; Mangard, S.; Kocher, P.; Genkin, D.; Yarom, Y.; Hamburg, M.: Meltdown: Reading Kernel Memory from User Space. In: USENIX Security Symposium. 2018.

[Li20a] Lipp, M.; Hadžić, V.; Schwarz, M.; Perais, A.; Maurice, C.; Gruss, D.: Take a Way: Exploring the Security Implications of AMD's Cache Way Predictors. In: AsiaCCS. 2020.

[Li20b] Lipp, M.; Schwarz, M.; Raab, L.; Lamster, L.; Aga, M. T.; Maurice, C.; Gruss, D.: Nethammer: Inducing Rowhammer Faults through Network Requests. In: SILM Workshop. 2020.

[Li21a] Lipp, M.: Exploiting Microarchitectural Optimizations from Software, Diss., Graz University of Technology, 2021.

[Li21b] Lipp, M.; Kogler, A.; Oswald, D.; Schwarz, M.; Easdon, C.; Canella, C.; Gruss, D.: PLATYPUS: Software-based Power Side-Channel Attacks on x86. In: IEEE S&P. 2021.

[Mo20] Moghimi, D.; Lipp, M.; Sunar, B.; Schwarz, M.: Medusa: Microarchitectural Data Leakage via Automated Attack Synthesis. In: USENIX Security Symposium. 2020.

[Mo98] Morgan, R.: Building an optimizing compiler. Digital Press, 1998.

[MOP08] Mangard, S.; Oswald, E.; Popp, T.: Power Analysis Attacks: Revealing the Secrets of Smart Cards. 2008.

[Ra21] Ragab, H.; Milburn, A.; Razavi, K.; Bos, H.; Giuffrida, C.: CROSSTALK: Speculative Data Leaks Across Cores Are Real. In: IEEE S&P. 2021.

[Sc18] Schwarz, M.; Lipp, M.; Gruss, D.; Weiser, S.; Maurice, C.-m.-t.-n.; Spreitzer, R.; Mangard, S.: KeyDrown: Eliminating Software-Based Keystroke Timing Side-Channel Attacks. In: NDSS. 2018.

[Sc19a] Schwarz, M.; Lipp, M.; Moghimi, D.; Van Bulck, J.; Stecklina, J.; Prescher, T.; Gruss, D.: ZombieLoad: Cross-Privilege-Boundary Data Sampling. In: ACM CCS. 2019.

[Sc19b] van Schaik, S.; Milburn, A.; Österlund, S.; Frigo, P.; Maisuradze, G.; Razavi, K.; Bos, H.; Giuffrida, C.: RIDL: Rogue In-flight Data Load. In: IEEE S&P. 2019.

[Sc20] van Schaik, S.; Minkin, M.; Kwong, A.; Genkin, D.; Yarom, Y.: CacheOut: Leaking Data on Intel CPUs via Cache Evictions. In: IEEE S&P. 2020.

[SHW11] Sorin, D. J.; Hill, M. D.; Wood, D. A.: A Primer on Memory Consistency and Cache Coherence. 2011.

[SL13] Shen, J. P.; Lipasti, M. H.: Modern Processor Design: Fundamentals of Superscalar Processors. Waveland Press, 2013.

[Sp17] Spreitzer, R.; Moonsamy, V.; Korak, T.; Mangard, S.: Systematic classification of side-channel attacks: a case study for mobile devices. IEEE Communications Surveys & Tutorials/, 2017.

[SP18] Stecklina, J.; Prescher, T.: LazyFP: Leaking FPU Register State using Microarchitectural Side-Channels. arXiv:1806.07480/, 2018.

[Va18] Van Bulck, J.; Minkin, M.; Weisse, O.; Genkin, D.; Kasikci, B.; Piessens, F.; Silberstein, M.; Wenisch, T. F.; Yarom, Y.; Strackx, R.: Foreshadow: Extracting the Keys to the Intel SGX Kingdom with Transient Out-of-Order Execution. In: USENIX Security Symposium. 2018.

[YF14] Yarom, Y.; Falkner, K.: Flush+Reload: a High Resolution, Low Noise, L3 Cache Side-Channel Attack. In: USENIX Security Symposium. 2014.

Moritz Lipp ist ein Sicherheitsforscher in Graz, Österreich. Seine Forschung beschäftigt sich vorrangig mit Mikroarchitekturangriffen. Er studierte an der Technischen Universität Graz, an der er auch sein Doktorat mit Auszeichnung zu diesem Thema abschloss. Moritz Lipp war Teil der Forschungsgruppen, die die Sicherheitslücken Meltdown, Spectre, Fallout, LVI, ZombieLoad und PLATYPUS entdeckten.

Restartstrategien[1]

Jan-Hendrik Lorenz[2]

Abstract: Restarts werden von vielen randomisierten Prozessen zur Leistungssteigerung verwendet. Wenn der Prozess nach einer bestimmten Zeit nicht erfolgreich war, wird er zurückgesetzt und ein neuer, unabhängiger Versuch gestartet. Allerdings gab es bisher nur wenig theoretische Untersuchungen zu Restarts. In dieser Dissertation [Lo21a] wird einerseits gezeigt, dass die Entscheidung, ob Restarts durchgeführt werden sollen, NP-hart ist, andererseits wird ein $(4+\varepsilon)$-Approximationsalgorithmus zur Bestimmung der optimalen Restartstrategie entwickelt. Darüber hinaus werden Methoden entwickelt, mit denen in den meisten Anwendungsfällen effizient entschieden werden kann, ob Restarts durchgeführt werden sollten und welches die optimale Restartstrategie ist. Diese Methoden werden in Verbindung mit Machine Learning eingesetzt, um einen SAT-Solver zu verbessern; dieser Solver konnte im Durchschnitt um mehr als das 50-fache beschleunigt werden. Wir untersuchen auch eine der am häufigsten verwendeten Restartstrategien und zeigen, dass eine wesentliche Eigenschaft, die für auf den natürlichen Zahlen definierten Prozesse gilt, nicht allgemeingültig ist.

1 Einleitung

Suchprozesse sind ein integraler Bestandteil des Alltagslebens. Auch die Wissenschaft hat sich mit verschiedenen Arten solcher Prozesse befasst. Dabei werden so vielfältige Themen wie das Aufspüren feindlicher U-Boote, das Auffinden von Schiffbrüchigen und die Nahrungssuche von Tieren untersucht. Es ist daher kaum verwunderlich, dass ein großer Aufwand betrieben wird, um besonders vielversprechende Suchstrategien zu ermitteln. In diesem Zusammenhang ist das Suchverhalten einer Reihe von Tierarten beobachtet und ein faszinierendes Muster festgestellt worden: Viele Arten tendieren zunächst zum systematischen Absuchen der unmittelbaren Umgebung; bleiben sie jedoch eine gewisse Zeit lang erfolglos, verlagern sie ihre Suche in ein anderes Areal, wo sie diese fortsetzen. Man kann das so interpretieren, dass nach einiger Zeit die aktuelle Suchumgebung als ungünstig eingestuft wird und ein anderes Gebiet vielversprechender ist. Fasst man dieses Suchverhalten als Algorithmus auf, so entspricht der Ortswechsel einem Zurücksetzen des Algorithmus, gefolgt von einem erneuten Versuch unter veränderten Ausgangsbedingungen.

Ein derartiges Zurücksetzen des Zustands ist ein essenzieller Bestandteil vieler Suchstrategien. Dabei haben insbesondere Luby et al. [LSZ93] wegweisende Pionierarbeit für eine bestimmte Form solcher Rücksetzungen geleistet, die als **Restarts** bekannt sind. Wird

[1] Englischer Titel der Dissertation: „Restart Strategies“

[2] Universität Ulm, Institut für Theoretische Informatik, James-Franck-Ring, 89069 Ulm, Deutschland, jan-hendrik.lorenz@alumni.uni-ulm.de

ein Restart durchgeführt, so wird der Zustand zurückgesetzt und darüber hinaus sind alle zukünftigen Schritte unabhängig von den vergangenen.

Motiviert durch die Arbeit von Luby et al. wurden die Auswirkungen von Restarts für eine Vielzahl von Problemstellungen untersucht. Dabei hat sich herausgestellt, dass Restarts häufig eine äußerst positive Wirkung auf die Leistung haben können. Diese Beobachtung stützt sich dabei vor allem auf experimentelle Untersuchungen. Theoretische Studien sind demgegenüber seltener, dennoch gibt es auch hier einige Resultate, die den Nutzen von Restarts bestätigen.

In der Algorithmik sind Neustarts zu einem elementaren Bestandteil von Algorithmen für eine Vielzahl von Problemen geworden. Dazu gehören unter anderem klassische Probleme wie das Problem des Handlungsreisenden [KSW17], das Erfüllbarkeitsproblem (SAT) [Hu07] und die gemischt-ganzzahlige Programmierung [FM14]. Darüber hinaus führen einige moderne Trainingsalgorithmen für neuronale Netze [LH17] ebenfalls Restarts durch, da dies die Qualität der trainierten Netze steigern kann. Neben der Algorithmik werden Neustarts auch in Netzwerkprotokollen wie TCP [Pa11] verwendet, da so eine zuverlässige Kommunikation gewährleistet wird. Weiterhin finden Restarts, ausgelöst durch ihre Relevanz in der Informatik, nun auch in anderen Disziplinen wie der Physik [EM11] und Biophysik [Ro16] erhebliche Beachtung. Wie man also sehen kann, können zahlreiche Bereiche innerhalb und außerhalb der Informatik von einem verbesserten Verständnis von Restarts profitieren.

2 Grundlagen

Um die Ergebnisse dieser Dissertation [Lo21a] einzuordnen, werden wir zunächst formell beschreiben, was Restartstrategien sind und zwei der gängigsten Restartstrategien vorstellen.

Eine **Restartstrategie** $\mathcal{T} = (t_1, t_2, \dots)$ ist, rein mathematisch gesehen, eine unendliche Folge. Jedes Folgenglied $t_i \in \mathcal{T}$ entstammt den positiven, erweiterten reellen Zahlen $t_i \in \mathbb{R}_+ \cup \{\infty\}$ und wird als **Restartzeit** bezeichnet. Die Anwendung einer Restartstrategie auf einen Algorithmus oder eine Prozedur führt zu einer semantischen Interpretation. Dies wird in Algorithmus 1 veranschaulicht.

Algorithmus 1 $\mathcal{A}_\mathcal{T}$: Eine modifizierte Version von $\mathcal{A}$ auf Basis der Restartstrategie $\mathcal{T}$.

Eingabe: Eingabe x und Restartstrategie $\mathcal{T} = (t_1, t_2, \dots)$.
1: **procedure** $A_\mathcal{T}(x)$
2: **for** $i \in \{1, 2, \dots\}$ **do**
3: Führe $\mathcal{A}(x)$ für t_i Zeiteinheiten aus. ▷ (Re-)start $\mathcal{A}$
4: **if** $\mathcal{A}(x)$ war erfolgreich **then**
5: Beende $A_\mathcal{T}$.

Zusammengefasst: Die Restartzeiten bestimmen, wie lange der Basisalgorithmus $\mathcal{A}$ läuft, wobei nach jedem Restart $\mathcal{A}$ neu initialisiert wird. Das bedeutet auch, dass eine unendliche

Restartzeit bedeutet, dass keine weiteren Restarts mehr durchgeführt werden, sondern der Basisalgorithmus $\mathcal{A}$ so lange läuft, bis er terminiert.

Nun stellt sich natürlich die Frage, welche Restartstrategie verwendet werden sollte. Zuvor ist aber zu klären, welches Maß für den Vergleich und die Bewertung verschiedener Restartstrategien zu verwenden ist. Zu diesem Zweck wird in der Regel der Erwartungswert herangezogen, da dieser das geeignetste Maß für die Beschreibung der langfristigen Durchschnittsleistung ist. Ziel ist es daher, Restartstrategien zu konstruieren, die die erwartete Laufzeit des jeweiligen Algorithmus minimieren. Zur Notation: Wir verwenden eine Zufallsvariable X, die die Laufzeit des betrachteten Algorithmus $\mathcal{A}$ auf der Eingabe x ohne Restarts darstellt. Wird nun die Restartstrategie $\mathcal{T}$ auf $\mathcal{A}(x)$ angewendet, so verwenden wir die Zufallsvariable $X_{\mathcal{T}}$, um die Laufzeit des modifizierten Algorithmus zu bezeichnen.

In diesem Rahmen konnten Luby et al. [LSZ93] in ihrer richtungsweisenden Arbeit die Frage beantworten, welche Restartstrategien verwendet werden sollten, indem sie zwei Strategien entwickelten. Die erste Strategie, die sogenannte **Fixed-Cutoff-Strategie**, besteht aus einer konstanten Restartzeit, es gilt also $\mathcal{T} = (t, t, t, \dots)$ für ein $t \in \mathbb{R}_+ \cup \{\infty\}$. Diese vermeintlich einfache Restartstrategie erweist sich als optimal.

Theorem 1 ([LSZ93]) *Für alle positiven Zufallsvariablen X eines Algorithmus $\mathcal{A}$ gibt es eine Fixed-Cutoff-Strategie $\mathcal{T} = (t, t, \dots)$ mit $t \in \mathbb{R}_+ \cup \{\infty\}$, sodass*

$$E[X_{\mathcal{T}}] \leq E[X_{\mathcal{L}}]$$

für alle Restartstrategien $\mathcal{L}$ gilt.

Trotz ihrer Optimalität hat die Fixed-Cutoff-Strategie einen offensichtlichen Schwachpunkt. Die optimale Fixed-Cutoff-Strategie unterscheidet sich für jeden Algorithmus und sogar für jede Eingabe. Mit anderen Worten: Die verwendeten Restartzeiten sind verschieden. **Universelle Restartstrategien** stellen eine Alternative dar. Dies sind Strategien, die weder vom Algorithmus noch von der Eingabe abhängen. Die wichtigste dieser Strategien ist **Lubys Strategie**. Die genaue Definition dieser Strategie kann der Arbeit von Luby et al. [LSZ93] entnommen werden. Vor allem Lubys Strategie kommt in vielen angewandten Algorithmen zum Einsatz. Der Hauptgrund dafür dürfte in einer äußerst attraktiven, theoretisch bewiesenen Eigenschaft liegen.

Theorem 2 ([LSZ93]) *Für alle auf den natürlichen Zahlen definierten Zufallsvariablen ist Lubys Strategie bis auf einen logarithmischen Faktor optimal.*

Außerdem gibt es keine universelle Strategie, die in jedem Fall weniger als einen logarithmischen Faktor von der optimalen Strategie entfernt ist. Man könnte also argumentieren, dass Lubys Strategie eine optimale universelle Restartstrategie ist.

3 Beitrag der Dissertation

Wie aus Theorem 1 hervorgeht, gibt es für jeden Prozess immer eine Fixed-Cutoff-Strategie, die in Bezug auf den betrachteten Erwartungswert optimal ist. Daraus ergeben sich jedoch gleich mehrere Fragen.

Zunächst einmal ist aus der Definition der Fixed-Cutoff-Strategie ersichtlich, dass die verwendete Restartzeit t den erweiterten reellen Zahlen $\mathbb{R}_+ \cup \{\infty\}$ entstammt. Dabei wird $t = \infty$ so interpretiert, dass keine Restarts durchgeführt werden. In diesem Fall gibt es keine Restartstrategie mit endlichen Restartzeiten, die die Leistung des betrachteten Prozesses verbessert. Im Gegenteil: Restarts können den Prozess so drastisch verschlechtern, dass er für alle praktischen Anwendungen faktisch unbrauchbar wird. Bevor man also eine Restartstrategie, wie etwa Lubys Strategie, implementiert, stellt sich die Frage, ob Restarts überhaupt sinnvoll sind, also ob sie die Leistung verbessern können. Trotz der offensichtlichen Relevanz für verschiedene Disziplinen wurde dieser Frage bisher wenig Aufmerksamkeit gewidmet.

Zweitens ist zwar bekannt, dass eine Fixed-Cutoff-Strategie optimal ist, für den konkreten Anwendungsfall muss jedoch geklärt werden, welche genau. Anders formuliert stellt sich also die Frage, welcher Restartzeitpunkt zur Optimierung des jeweiligen Prozesses herangezogen werden sollte. Auch diese Frage wurde bisher nicht geklärt. Drittens macht Theorem 1 keine Aussage darüber, in welchem Maße die optimale Restartstrategie die Leistung eines Prozesses verbessern kann. Auch diese Fragestellung ist von zentraler Bedeutung.

Diese drei Aspekte wurden teilweise im Kontext einzelner Algorithmen betrachtet, beispielsweise im Zusammenhang mit bestimmten SAT-Algorithmen (z. B. [Hu07]). Ein einheitlicher Ansatz zur Beurteilung dieser Fragen fehlte jedoch bisher. Insbesondere mussten die Methoden zur Beurteilung, ob sich eine Restartstrategie lohnt und wenn ja, welche, für praktisch jede Problemstellung neu erarbeitet werden. Ziel sollte es daher sein, solche Methoden so zu entwickeln, dass sie in Zukunft auf die meisten Problemstellungen angewendet werden können, sowohl für theoretische Untersuchungen wie Komplexitätsanalysen als auch für praktische Anwendungen. Dies ist eines der Hauptziele der vorliegenden Dissertation [Lo21a] und wird in den Abschnitten 3.1 bis 3.3 aus verschiedenen Blickwinkeln eingehender erörtert.

Ein zweiter wesentlicher Schwerpunkt ist die Untersuchung der theoretischen Eigenschaften von Lubys Strategie. Wie aus Theorem 2 hervorgeht, ist Lubys Strategie hinsichtlich ihrer Leistung nicht allzu weit von der optimalen Restartstrategie entfernt, zumindest wenn der betrachtete Prozess auf den natürlichen Zahlen definiert ist. Es stellt sich also die Frage, ob Theorem 2 auch unter anderen Bedingungen gültig ist, zum Beispiel, wenn auf den reellen Zahlen gearbeitet wird. Dieser Frage soll in Sektion 3.4 nachgegangen werden.

3.1 Über die Komplexität von Restarts

Wenn man vor der Frage steht, ob Restarts für einen bestimmten Prozess, etwa einen Algorithmus oder ein Netzwerkprotokoll, verwendet werden sollten, wäre es wünschenswert, über geeignete Algorithmen zu verfügen, die bei der Entscheidungsfindung helfen. Solche Algorithmen sollten idealerweise eine allgemeine Beschreibung der Prozesseigenschaften als Eingabe annehmen und als Antwort liefern, ob der Prozess durch Restarts optimiert werden kann, sowie die optimale Restartstrategie ermitteln. Darüber hinaus sollten solche Algorithmen auch effizient sein. Die Frage, welche Eigenschaften solche Algorithmen haben, wird aus der Perspektive der Komplexitätstheorie betrachtet. Bislang wurden Restarts nicht unter diesem Gesichtspunkt betrachtet.

Dazu muss zunächst klar definiert werden, wie die Eingabe für solche Algorithmen aussehen soll. Da Restarts nur für solche Prozesse sinnvoll sein können, die zumindest partiell dem Zufall unterliegen[3], kann man sich hier an den Werkzeugen der Stochastik bedienen. Besonders bekannt sind hierbei die Wahrscheinlichkeitsfunktion und die Verteilungsfunktion, mit denen alle (diskreten) Prozesse beschrieben werden können. Diese beiden Funktionstypen sind also geeignete Kandidaten für unsere Zwecke, aber es muss noch geklärt werden, *wie* man sie als Eingabe für einen Algorithmus bereitstellt. Zu diesem Zweck eignen sich Straight-Line-Programme. Dabei handelt es sich um Beschreibungen, wie die Werte der Wahrscheinlichkeits- beziehungsweise Verteilungsfunktion zu berechnen sind.

Nun kann man sich der Komplexität der eigentlichen Probleme zuwenden. Zu prüfen, ob ein Prozess durch Restarts optimiert werden kann, erweist sich als NP-schwer [Lo19], unabhängig davon, ob die mit dem Prozess verbundene Wahrscheinlichkeits- oder Verteilungsfunktion herangezogen wird. Sollte, wie allgemein angenommen, $\mathrm{P} \neq \mathrm{NP}$ gelten, dann impliziert dies, dass es nicht möglich ist, die optimale Restartstrategie effizient zu ermitteln.

Im vorliegenden Fall wäre es jedoch oftmals ausreichend, wenn eine hinreichend gute Restartstrategie gefunden werden könnte. Im Idealfall gäbe es einen effizienten Algorithmus, der eine Restartstrategie erzeugt, deren zugehöriger Erwartungswert garantiert höchstens um einen konstanten Faktor c höher ist als der Erwartungswert bei Verwendung der optimalen Strategie. Algorithmen dieser Art werden als c-Approximationsalgorithmen bezeichnet.

Nun stellt sich heraus, dass es keinen effizienten c-Approximationsalgorithmus für den Fall gibt, in dem die Wahrscheinlichkeitsfunktion gegeben ist, unabhängig davon, wie c gewählt wird. Für die Verteilungsfunktion ergibt sich dagegen ein anderes Bild: Für diesen Fall konnte ein effizienter $(4+\varepsilon)$-Approximationsalgorithmus entwickelt werden [LS20]. Interessanterweise erzeugt der Algorithmus zwar eine Restartstrategie, deren zugehöriger Erwartungswert höchstens um einen Faktor $(4+\varepsilon)$ höher ist als der Erwartungswert der optimalen Strategie, aber die Bestimmung dieses assoziierten Erwartungswerts erweist

[3] Es gibt noch andere Arten von Restarts, die auch für deterministische Prozesse nützlich sein können. Diese werden hier jedoch nicht behandelt.

sich wiederum als nicht praktikabel. Konkret bedeutet dies, dass die Berechnung des Erwartungswerts eines Prozesses mit einer gegebenen Restartstrategie #P-schwer ist [Lo19].

3.2 Methoden zur Bestimmung der optimalen Restartstrategie

Die Ergebnisse aus Sektion 3.1 zeigen, dass es im Allgemeinen anspruchsvoll ist, die optimale Restartstrategie zu bestimmen oder überhaupt eine Aussage darüber zu treffen, ob es eine Restartstrategie gibt, die die Leistung des betrachteten Prozesses verbessert. Ziel dieses Abschnitts ist es, einen Ansatz herzuleiten, mit dem viele Prozesse analysiert werden können, sodass die NP-Schwere in der Praxis meist unerheblich ist. In diesem Zusammenhang hat Wolter [Wo10] bereits eine Bedingung entwickelt, mit der analysiert werden kann, ob Restarts zur Leistungssteigerung eingesetzt werden können. Die in diesem Abschnitt vorgestellten Ergebnisse basieren größtenteils auf [Lo18].

Es ist uns gelungen, Verfahren zu entwickeln, mit denen festgestellt werden kann, ob Neustarts zur Leistungssteigerung eingesetzt werden können, und falls ja, mit denen die optimale Restartstrategie ermittelt werden kann. Diese entwickelten Methoden bieten zwei wesentliche Vorteile gegenüber dem Ansatz von Wolter. Erstens verwenden wir die sogenannte Quantilfunktion, die auf dem Intervall $(0, 1)$ definiert ist. Diese Eigenschaft kann algorithmisch zur Auswertung der notwendigen Kriterien genutzt werden. Zweitens erfordern die Bedingungen von Wolter die Kenntnis des sogenannten rechten Randes der betrachteten Verteilungen. Kennt man die genaue Verteilungsfunktion nicht, sondern approximiert sie auf der Basis von Stichproben, würde dies bedeuten, dass man in besonderem Maße von den Extremwerten beziehungsweise Ausreißern der Stichprobe abhängig ist. Dies bedeutet zum einen, dass die Erhebung der Stichproben sehr lange dauern kann, da die Experimente nicht vorzeitig abgebrochen werden sollten, und zum anderen, dass zu kleine Stichproben die Ergebnisse stark verfälschen können. Die von uns entwickelten Ansätze haben dieses Problem jedoch nicht.

Um die Anwendbarkeit der entwickelten Verfahren zu demonstrieren, werden einige Verteilungstypen analysiert. Insbesondere konnte gezeigt werden, dass sich Restarts für alle Verteilungen lohnen, die die sogenannte Long-Tail-Eigenschaft haben [WL21]. Dies ist eine erhebliche Verbesserung des zuvor bekannten Ergebnisses von Gomes et al. [GSK98], die diese Aussage für Potenzgesetzverteilungen zeigten. Um zu verdeutlichen: Jede Potenzgesetzverteilung hat die Long-Tail-Eigenschaft. Es gibt jedoch einige in der Praxis bedeutsame Verteilungstypen, wie etwa die Lognormal- und manche Weibullverteilungen, die zwar die Long-Tail-Eigenschaft haben, aber keine Potenzgesetzverteilungen sind.

Darüber hinaus gab es bislang keine systematischen Untersuchungen über die optimale Restartstrategie, d. h. für welche Wahrscheinlichkeitsverteilungen welche Fixed-Cutoff-Strategie optimal ist. Diese Lücke wird in der Dissertation geschlossen. Das hergeleitete Verfahren wird angewandt, um einige der bedeutendsten Wahrscheinlichkeitsverteilungen

zu analysieren. Diese Ergebnisse erleichtern die Prozessanalyse in Bezug auf die Frage, ob Restarts durchgeführt werden sollten und welche Strategie optimal ist, erheblich.

3.3 Optimale Restarts durch Machine Learning

Um die optimale Restartstrategie in der Praxis bestimmen zu können, muss die entsprechende Wahrscheinlichkeitsverteilung im Voraus bekannt sein. Im Falle eines randomisierten Algorithmus wäre dies beispielsweise die Wahrscheinlichkeitsverteilung, die das Verhalten des Algorithmus für eine bestimmte Probleminstanz beschreibt. Dies ist jedoch in der Regel nicht der Fall. Zwar lässt sich die Wahrscheinlichkeitsverteilung durch wiederholtes Lösen der betreffenden Instanz empirisch annähern, doch in den meisten praktischen Anwendungen sind die Instanzen nicht mehr von Interesse, sobald sie einmal gelöst worden sind.

Das Ziel dieses Abschnitts ist es, dieses Problem mit Hilfe von Machine Learning zu lösen. Zur Illustration ziehen wir zu diesem Zweck den SAT-Solver probSAT [BS16] heran. Es gibt mehrere Aspekte, die probSAT für eine solche Untersuchung attraktiv machen. Erstens ist es ein sehr leistungsstarker Algorithmus; Implementierungen von probSAT gewannen die entsprechenden Disziplinen bei den SAT-Competions 2013, 2014 und 2017. Außerdem verwenden andere, ähnlich erfolgreiche Solver probSAT als eine ihrer Kernkomponenten. Man könnte also annehmen, dass probSAT bereits bis in die kleinsten Details optimiert worden ist. Bemerkenswert ist dabei, dass probSAT keine Restartstrategie verwendet.

Für den Machine-Learning-Ansatz gehen wir in mehreren Phasen vor. In den Vorbereitungsphasen wird probSAT auf einer großen Sammlung von Instanzen wiederholt ausgeführt, sodass für jede dieser Instanzen die Wahrscheinlichkeitsverteilung, die die Laufzeit von probSAT beschreibt, approximiert werden kann. Es stellt sich heraus, dass diese Verteilungen aus einer Familie von parametrisierten Wahrscheinlichkeitsverteilungen stammen. Durch diese Erkenntnis kann man das Laufzeitverhalten von probSAT für jede Instanz durch drei Parameter beschreiben.

Es folgte die Trainingsphase. Als Label wurden hier die drei Parameter verwendet, die das Laufzeitverhalten beschreiben. Als Merkmale wurden typische Merkmale von SAT-Instanzen wie die Anzahl der Variablen und Klauseln verwendet. Zudem wurden aber auch andere Algorithmen kurz auf der entsprechenden Instanz ausgeführt. Das Verhalten dieser Algorithmen, wie in etwa die maximale Anzahl der erfüllten Klauseln innerhalb des kurzen Durchlaufs, wurde ebenfalls als Merkmal verwendet.

Das bedeutet, dass die Wahrscheinlichkeitsverteilung für ungesehene Instanzen vorhergesagt werden kann, indem zunächst die Merkmale dieser Instanz bestimmt werden und diese dann durch den Machine-Learning-Ansatz zur Bestimmung der drei Parameter herangezogen werden. Diese drei Parameter definieren eine eindeutige Wahrscheinlichkeitsverteilung, anhand derer dann entschieden werden kann, ob Restarts durchgeführt werden sollten und wenn ja, welches die optimale Restartstrategie ist, die dann von probSAT verwendet wird.

Hier zeigt sich erneut der Vorzug der Methoden aus Sektion 3.2, da so diese Berechnungen algorithmisch automatisiert werden können.

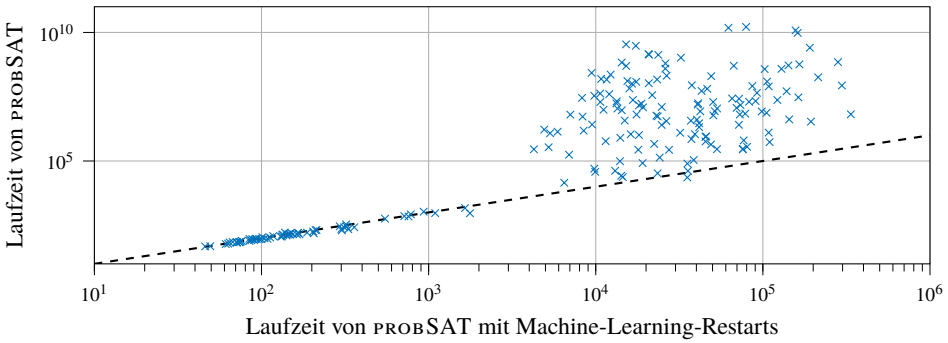


Abb. 1: Jede Markierung entspricht der logarithmisch skalierten durchschnittlichen Laufzeit einer einzelnen Instanz. Die durchschnittlichen Laufzeiten der Machine-Learning-Restart-Strategie sind auf der x-Achse dargestellt, die der Originalversion von probSAT auf der y-Achse. Die modifizierte Version ist effizienter, wenn der Marker oberhalb der gestrichelten Linie liegt.

Um diesen Ansatz zu evaluieren, wird die so modifizierte Version von probSAT mit der unmodifizierten Version verglichen. Dazu werden Instanzen verwendet, die den Konditionen der SAT-Competition entsprechen. Die modifizierte Version erreicht im Vergleich zur unmodifizierten Version einen durchschnittlichen Speedup-Faktor von mehr als 50. Wie zudem aus Abb. 1 ersichtlich ist, erfolgt diese Verbesserung vor allem bei besonders schweren Instanzen, bei denen oft eine Leistungssteigerung um mehrere Größenordnungen erreicht werden kann. Dieses Ergebnis wird auch in einer separaten Auswertung auf den Instanzen und unter den Bedingungen der SAT-Competition bestätigt. Somit wurde gezeigt, dass die modifizierte Version von probSAT dem heutigen State-of-the-Art entspricht.

Diese Ergebnisse belegen, dass die Bedeutung von Restarts in praktischen Anwendungen nach wie vor unterschätzt wird. Außerdem lässt sich der hier vorgestellte Ansatz ohne Weiteres auf andere Probleme anwenden. Die einzige Herausforderung ist hier die Suche nach informativen Merkmalen zur Bestimmung der Wahrscheinlichkeitsverteilung.

3.4 Über die Eigenschaften von Lubys Strategie

In vielen Anwendungen wird Lubys Strategie verwendet. Erstens erspart sie die Suche nach der optimalen Restartstrategie und zweitens garantiert Theorem 2 scheinbar, dass die Leistung nicht zu weit von der der optimalen Strategie abweicht. Die Aussage von Theorem 2 ist jedoch auf Prozesse beschränkt, die auf den natürlichen Zahlen definiert sind. In zahlreichen Situationen arbeitet man jedoch mit reellen Zahlen, zum Beispiel wenn die Leistung eines Algorithmus in Echtzeit gemessen wird. Daher ist es wichtig zu wissen, ob die Beschränkung auf die natürlichen Zahlen nur eine rein technische Annahme zur

Vereinfachung des Beweises ist oder ob die Aussage von Theorem 2 nicht generell gilt. Die in der Dissertation vorgestellten Ergebnisse beruhen dabei auf [Lo21b].

Es konnte bewiesen werden, dass Theorem 2 im Allgemeinen nicht gilt. Tatsächlich gibt es keine universelle Restartstrategie, die im Allgemeinen eine Schranke in der Form von Theorem 2 zulässt. Dabei ist auch der logarithmische Faktor nicht der begrenzende Faktor, man kann ihn durch jeden anderen Faktor ersetzen und dennoch gibt es keine universelle Restartstrategie mit einer entsprechenden Leistungsgarantie.

3.5 Weitere Beiträge

Aus Platzgründen konnte nicht auf alle Beiträge der Dissertation im Detail eingegangen werden. Zwei wesentliche Bereiche sollen an dieser Stelle gestreift werden. In Sektion 3.1 wurde für die Komplexitätsanalysen angenommen, dass die übergebenen Straight-Line-Programme tatsächlich eine Wahrscheinlichkeits- bzw. Verteilungsfunktion beschreiben. Dies entspricht einem sogenannten Promise-Problem. Es wurde auch die Komplexität dieser Promises betrachtet. Dabei zeigt sich, dass die Entscheidung, ob ein gegebenes Straight-Line-Programm einer Verteilungsfunktion entspricht, coNP-schwer ist [LS20]. Die Verifikation einer Wahrscheinlichkeitsfunktion ist sogar $P^{\#P}$-schwer [LS20].

Meistens werden Restarts verwendet, um einen Prozess im Hinblick auf den assoziierten Erwartungswert zu optimieren. Manchmal werden sie aber auch eingesetzt, um die Wahrscheinlichkeit zu maximieren, dass eine Deadline eingehalten wird. Dieses Szenario wird in der Dissertation systematisch analysiert, mit der zusätzlichen Erweiterung, dass der Prozess parallel auf mehreren Prozessoren läuft. Es werden Bedingungen für das Finden der optimalen Restart-Strategie hergeleitet [Lo16]. Es wird auch gezeigt, dass man in diesem Szenario superlinear von der Anzahl der eingesetzten Prozessoren profitiert [Lo16].

Literatur

[BS16] Balint, A.; Schöning, U.: Engineering a Lightweight and Efficient Local Search SAT Solver. In: Algorithm Engineering. Springer, S. 1–18, 2016.

[EM11] Evans, M. R.; Majumdar, S. N.: Diffusion with Stochastic Resetting. Physical Review Letters 106/16, S. 160601, 2011.

[FM14] Fischetti, M.; Monaci, M.: Exploiting Erraticism in Search. Operations Research 62/1, S. 114–122, 2014.

[GSK98] Gomes, C. P.; Selman, B.; Kautz, H.: Boosting Combinatorial Search Through Randomization. In: AAAI. American Association for Artificial Intelligence, S. 431–437, 1998.

[Hu07] Huang, J.: The Effect of Restarts on the Efficiency of Clause Learning. In: IJCAI. Morgan Kaufmann Publishers Inc., S. 2318–2323, 2007.

[KSW17] Kadioglu, S.; Sellmann, M.; Wagner, M.: Learning a Reactive Restart Strategy to Improve Stochastic Search. In: LION. Springer, S. 109–123, 2017.

[LH17] Loshchilov, I.; Hutter, F.: SGDR: Stochastic Gradient Descent with Warm Restarts. In: ICLR, conference track. 2017.

[Lo16] Lorenz, J.-H.: Completion Probabilities and Parallel Restart Strategies under an Imposed Deadline. PLOS ONE 11/10, S. 1–15, 2016.

[Lo18] Lorenz, J.-H.: Runtime Distributions and Criteria for Restarts. In: SOFSEM. Springer, S. 493–507, 2018.

[Lo19] Lorenz, J.-H.: On the Complexity of Restarting. In: CSR. Springer, S. 250–261, 2019.

[Lo21a] Lorenz, J.-H.: Restart Strategies, Dissertation, Ulm University, 2021.

[Lo21b] Lorenz, J.-H.: Restart Strategies in a Continuous Setting. Theory of Computing Systems 65/8, S. 1143–1164, 2021.

[LS20] Lorenz, J.-H.; Schöning, U.: Promise Problems on Probability Distributions. In: Complexity and Approximation. Springer, S. 57–66, 2020.

[LSZ93] Luby, M.; Sinclair, A.; Zuckerman, D.: Optimal Speedup of Las Vegas Algorithms. Information Processing Letters 47/4, S. 173–180, 1993.

[Pa11] Paxson, V.; Allman, M.; Chu, J.; Sargent, M.: Computing TCP's retransmission timer, Techn. Ber., 2011.

[Ro16] Roldán, É.; Lisica, A.; Sánchez-Taltavull, D.; Grill, S. W.: Stochastic resetting in backtrack recovery by RNA polymerases. Physical Review E 93/6, S. 062411, 2016.

[WL21] Wörz, F.; Lorenz, J.-H.: Evidence for Long-Tails in SLS Algorithms. In: ESA. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 82:1–82:16, 2021.

[Wo10] Wolter, K.: Stochastic Models for Fault Tolerance: Restart, Rejuvenation and Checkpointing. Springer Science & Business Media, 2010.

Jan-Hendrik Lorenz absolvierte sein Bachelor- und Masterstudium in Informatik an der Universität Ulm. Sein Promotionsstudium über Restartstrategien führte er am Institut für Theoretische Informatik der Universität Ulm unter der Betreuung von Prof. Dr. Uwe Schöning durch. Jan-Hendrik Lorenz erhielt für seine Veröffentlichungen bei der SOFSEM 2018, der CSR 2019 sowie der ESA 2021 jeweils den Best Student Paper Award. Seine umfangreichen Forschungsinteressen kommen auch in akademischen Publikationen in den Bereichen der Blockchain-Technologie, Spieltheorie, Komplexitätstheorie und Algorithm Design zum Ausdruck.

Verbesserung der Praxistauglichkeit der statischen Taint-Analyse [1]

Linghui Luo[2]

Abstract: Statische Taint-Analyse ist eine Programmanalysetechnik, die bösartige Software aufspüren und ein breites Spektrum von Sicherheitslücken aufdecken kann. Obwohl sowohl in der Industrie als auch im akademischen Bereich viele statische Taint-Analyse-Werkzeuge entwickelt wurden, werden nur sehr wenige davon in der Industrie eingesetzt. Diese Kurzfassung stellt meine Dissertation vor, in der ich die Gründe für die mangelnde Nutzung von statische Taint-Analyse-Werkzeuge in der Praxis untersuchte. Ich entwickelte verschiedene Ansätze, die drei erkannte Probleme angehen, um die Praxistauglichkeit der statischen Taint-Analyse zu verbessern.

1 Einleitung

Sicherheitspannen kommen täglich vor und stellen eine ernsthafte Bedrohung für Unternehmen dar. Wie im IBM-Bericht "2019 Cost of a Data Breach Report" angegeben, belaufen sich die durchschnittlichen Gesamtkosten einer Datenschutzverletzung auf 3,91 Millionen US-Dollar. Allein im Jahr 2019 gab es in den USA 1473 Datenschutzverletzungen, bei denen mehr als 164 Millionen Datensätze gestohlen wurden [St20]. Im Mai 2021 musste der Treibstoffpipeline-Betreiber Colonial Pipeline 75 Bitcoins (fast 5 Millionen US-Dollar) zahlen, um seine von einer Ransomware gestohlenen Daten wiederzuerlangen [MK21]. Seit der ersten Veröffentlichung der deutschen Luca-App zur Verfolgung von Kontakten während der COVID-19-Pandemie wurde eine Kette von Sicherheitslücken entdeckt, und es entstand eine große Skepsis gegenüber ihrer Nützlichkeit [Re21][Ke21]. Solche Vorfälle kosten nicht nur viel Geld und schaden dem Ruf des Unternehmens, sondern können auch eine Gefahr für die nationale Sicherheit darstellen. Wie 2019 berichtet, wurden Bundeskanzlerin Angela Merkel und Hunderte von deutschen Politikern von einem Datenhack getroffen, bei dem vertrauliche Briefe, Kontaktinformationen und Parteimemos auf Twitter durchsickerten [Me19]. Die Gewährleistung der Sicherheit von Softwareanwendungen ist in der heutigen schnelllebigen technologischen Welt wichtiger denn je.

Es gibt zwar viele Techniken, die gewährleisten, dass Softwareanwendungen sicher und widerstandsfähig gegen Cyberangriffe sind, dennoch bleiben Angriffe aufgrund der begrenzten Wirksamkeit und Akzeptanz dieser Techniken eine große Bedrohung. In dieser Dissertation [Lu21] konzentrieren wir uns auf eine Technik, die statische Taint-Analyse

[1] Englischer Titel der Dissertation: "Improving Real-World Applicability of Static Taint Analysis"

[2] Universität Paderborn, Fakultät für Elektrotechnik, Informatik u. Mathematik, Warburger Str. 100, 33098 Paderborn, Deutschland. linghui@outlook.de

heißt. Sie analysiert Programme, ohne sie auszuführen, und ihre Anwendbarkeit reicht von der Offenlegung undefinierter Verhaltensweisen (z.B. Null-Pointer-Dereferenzen) über die Erkennung von Sicherheitslücken (z.B. SQL-Injections) bis hin zur Verhinderung von Malware-Infektionen (z.B. Datendiebstahl). Insbesondere die statische Taint-Analyse ist in der Lage, eine Vielzahl von Sicherheitslücken und bösartigen Verhaltensweisen in Softwareanwendungen zu erkennen. Sie verfolgt Datenflüsse von sensiblen Quellen (z.B. Programmierschnittstelle (API), die nicht vertrauenswürdige Benutzereingaben oder private Daten liest) zu sensiblen Senken (z.B. eine API, die eine gefährliche Funktion ausführt oder Daten ins Internet stellt). Solche Datenflüsse werden als Taint-Flows bezeichnet. Viele bekannte Sicherheitsschwachstellen können durch Taint-Flows ausgelöst werden, z.B. Datendiebstahl, SQL-Injections, Cross-Site Scripting usw.

2 Problemstellung

In der Vergangenheit wurden sowohl in der Industrie als auch im akademischen Bereich viele statische Taint-Analyse-Werkzeuge entwickelt, aber nur wenige von ihnen haben in der Industrie eine breite Anwendung gefunden. Dafür gibt es zwei gegenläufige Gründe: Skalierbarkeit einerseits und gute Ergebnisse andererseits. Statische Werkzeuge gelten als gut, wenn sie beim Benchmarking eine hohe Genauigkeit (d. h. eine niedrige Rate an falsch-positiven Ergebnissen) und eine hohe Trefferquote (d. h. eine niedrige Rate an falsch-negativen Ergebnissen) liefern. Gute Ergebnisse zu erzielen und gleichzeitig skalierbar zu sein, ist jedoch bei realen Softwareanwendungen aufgrund ihrer Größe und Komplexität eine große Herausforderung. Viele Taint-Analyse-Werkzeuge haben ihre Genauigkeit und Trefferquote anhand von Mikro-Benchmark-Anwendungen (kleine Programme mit künstlich konstruierten Schwachstellen) bewertet und wurden daher optimiert, um gute Ergebnisse auf diesen zu erzielen. Im Gegensatz zur Evaluierung von Mikro-Benchmark-Anwendungen sind Evaluierungen von realen Softwareanwendungen (große und komplexe Programme) aufgrund des Mangels an etablierten Benchmarks unüblich. Dies führt zu drei Problemen, die die Anwendbarkeit dieser Analysewerkzeuge in der Praxis beeinträchtigen:

- **Übliche Benchmarks sind klein und unvollständig**: Mikro-Benchmark-Anwendungen nutzen selten die gesamte Bandbreite realer Plattformen, was dazu führt, dass Analysen aufgrund unvollständiger Modellierung Probleme übersehen [BGC15].
- **Probleme, über die Werkzeuge warnen, sind oft von begrenztem Interesse**: Analysewerkzeuge ignorieren reale Szenarien, wie z.B. Code, der aus Gründen der Skalierbarkeit an Plattformversionen gebunden ist [Li17]. Dies kann zu Warnungen führen, die für einen bestimmten Benutzer uninteressant oder sogar unrealisierbar sind, d. h. zur Laufzeit niemals auftreten können [Ar15].
- **Geringe Akzeptanz bei Entwicklern**: Mikro-Benchmark-Anwendungen sind in der Regel einfach gehalten, aber echte Programme sind sehr umfangreich und enthalten

oft komplizierte Probleme. Entwickler brauchen Unterstützung durch Werkzeuge, um diese schwierigen Probleme in der Praxis zu verstehen. Statische Analysewerkzeuge, die in der Wissenschaft entwickelt wurden, beschränken sich meist auf automatisierte Experimente, bei denen die Analysen als Befehlszeilenwerkzeuge ausgeführt werden, wobei den Aspekten der Benutzerfreundlichkeit auf Seiten der Entwickler wenig bis gar keine Beachtung geschenkt wird. Die Integration dieser Analysen in Integrierte Entwicklungsumgebungen (IDEs), die üblicherweise von Entwicklern verwendet werden, war weniger ausgeprägt.

In dieser Dissertation gehen wir diese drei Probleme an, um die Praxistauglichkeit der statischen Taint-Analyse zu verbessern. Im Folgenden werden die jeweiligen Beiträge zu den einzelnen Problemen vorgestellt.

3 Übliche Benchmarks sind klein und unvollständig

Um das erste Problem anzugehen, ist der erste Schritt, realistischere Benchmarks zu konstruieren. Als ersten Beitrag dieser Dissertation haben wir eine realistische Malware-Benchmark-Suite für die Android-Taint-Analyse erstellt. Wir konzentrieren uns in dieser Arbeit auf Android, aber ähnliche Probleme treten auch in anderen Umgebungen auf [An20; Jo13; Sr11; WR13]. Diese Benchmark-Suite, genannt TaintBench, ist die erste realistische Suite in diesem Bereich mit einer dokumentierten Ground-Truth. Bisher fehlen realistischen Benchmarks fast immer die Ground-Truth, die aber für eine reproduzierbare Evaluierung benötigt wird. Unsere TaintBench Benchmark-Suite enthält 39 Malware-Anwendungen mit 249 dokumentierten Benchmark-Fällen. Zusammen mit der Suite haben wir eine Reihe von Hilfswerkzeugen entwickelt, die eine schnellere Erstellung von Benchmark-Suiten, eine reproduzierbare Evaluierung von statischen Taint-Analysewerkzeugen mit dieser Suite und eine einfachere Bewertung statischer Ergebnisse ermöglichen. Mit TaintBench haben wir populäre statische Taint-Analyse-Werkzeuge (nämlich FlowDroid [Ar14] und Amandroid [We14]) evaluiert. Zum Vergleich haben wir dieselben Analysewerkzeuge mit der Mikro-Benchmark-Suite DroidBench [Dr16] evaluiert. Unsere Ergebnisse zeigen, dass diese Analysewerkzeuge bei reeller Malware im Vergleich zu Mikro-Benchmark-Anwendungen eine wesentlich niedrigere Genauigkeit und Trefferquote aufweisen. Selbst bei einer unrealistischen Konfiguration dieser Analysewerkzeuge (d. h. sie werden explizit mit Quellen und Senken konfiguriert, die von den bösartigen Datenflüssen verwendet werden) bleibt die Mehrheit der bösartigen Datenflüsse in TaintBench unerkannt. Wir haben das Analysewerkzeug FlowDroid weiter untersucht, das in unserer Evaluierung das beste Ergebnis mit TaintBench erzielte. Unsere Untersuchung ergab, dass 35% der bösartigen Datenflüsse in TaintBench von FlowDroid nicht erkannt werden konnten, weil relevante Methoden in den Call-Graphen fehlten. Es wird deutlich, dass man bessere Call-Graphen konstruieren muss.

Unser nächster Beitrag GENCG ist ein neuer Ansatz, einen Call-Graphen zu erstellen, der vollständiger, aber dennoch überschaubar ist. Call-Graphen sind wichtige Bausteine für interprozedurale statische Analysen, die für moderne Framework-basierte Anwendungen schwierig zu erstellen sind. Um skalierbar zu sein, modellieren die meisten statischen Analysewerkzeuge die Effekte der Frameworks, anstatt sie zu analysieren [Ar14; Go15; Sr11; We14]. FlowDroid zum Beispiel modelliert das Verhalten des Android-Frameworks, indem es eine Dummy-Main-Methode konstruiert, die den Lebenszyklus jeder Android-Komponente simuliert und die Analyse von dort aus startet. Diese präzise Modellierung ist jedoch schwer aufrechtzuerhalten, da jedes Jahr eine neue Version von Android mit neuen APIs herauskommt, die neue Verhaltensweisen in das Framework einführen. Außerdem ist es unpraktisch, dies für jedes Framework zu tun. Beliebte Frameworks für Java-Enterprise Anwendungen wie Spring verwenden Annotationen, die das Framework benutzt um zu entscheiden, welcher Code durch Reflexion ausgeführt werden soll. Um nützlich zu sein, muss ein Call-Graph diese Frameworks berücksichtigen: Sie zu analysieren ist unpraktisch, aber die Modellierung ist es auch.

In dieser Dissertation schlagen wir einen allgemeinen Ansatz zur Modellierung von Java-Frameworks vor. Unser Ansatz GENCG ist nicht auf ein bestimmtes Framework oder Analysewerkzeug beschränkt und daher in hohem Maße wiederverwendbar. In GENCG haben wir das Werkzeug AVERROES-GENCG entwickelt - eine Verbesserung von AVERROES [AL13], welches eine Platzhalterbibliothek für ein bestimmtes Android/Java-Framework erzeugt. Dieser generierte Platzhalter kann durch gängige Algorithmen zur Konstruktion von Call-Graphen (z.B. VTA, RTA, Spark [LH03; Su00]) und von Taint-Analyse als Ersatz für das ursprüngliche Framework verwendet werden. Das Verhalten des Frameworks wird im Code des Platzhalters modelliert und spiegelt sich in den konstruierten Call-Graphen wider. Während eine generische Approximation ungenau sein kann, zeigen wir, dass unsere sorgfältig konstruierte Approximation gut funktioniert. Wir demonstrieren ihre Verallgemeinerung mit zwei Frameworks - Android und Spring. Experimente auf Android mit einer Taint-Analyse zeigen, dass unser Ansatz vollständigere Call-Graphen erzeugt als die ursprüngliche Analyse. Infolgedessen werden sowohl die Genauigkeit (von 0,83 auf 0,88) als auch die Trefferquote (von 0,20 auf 0,32) der Taint-Analyse auf TAINTBENCH verbessert. Um die Allgemeingültigkeit zu demonstrieren, stellen wir vor, wie unser Ansatz auf Webanwendungen mit dem Spring-Framework angewendet werden kann. Die Evaluierung mit einer von uns erstellten Mikro-Benchmark-Suite von 42 Spring-basierten Webanwendungen zeigt vielversprechende Ergebnisse.

4 Probleme, über die Werkzeuge warnen, sind oft von begrenztem Interesse

Selbst bei einem hinreichend vollständigen Call-Graph ist die Genauigkeit der Analyse entscheidend. Um die Genauigkeit zu verbessern, werden von statischen Taint-Analyse-Werkzeugen verschiedene Sensitivitäten berücksichtigt. Um Aliasing und Virtual-Dispatch

in Java-Programmen zu behandeln, wenden statische Analyzewerkzeug Kontext-, Objekt- und Feldsensitivitäten an. Während eine flusssensitive Analyse die Reihenfolge der Anweisungen berücksichtigt, werden bei einer pfadsensitiven Analyse Verzweigungsbedingungen ausgewertet. Obwohl die meisten statischen Taint-Analysewerkzeuge mehrere Sensitivitäten gleichzeitig unterstützen, wird die Pfadsensitivität in der Regel ausgelassen. Dies führt zu Warnungen, die für Entwickler uninteressant sind, oder sogar zu falsch-positiven Ergebnissen. Der dritte Beitrag dieser Dissertation befasst sich mit diesem Problem. Wir haben einen Ansatz entwickelt, der partielle Pfadeinschränkungen berechnet, um die Ergebnisse einer Taint-Analyse zu verbessern. Wir haben ein Analysewerkzeug namens COVA implementiert, das die Datenflussanalyse mit der Nutzung des Satisfiability-Modulo-Theories-Solver Z3 [MB08] kombiniert. COVA kann so konfiguriert werden, dass es Informationen verfolgt, an denen man interessiert ist, z.B. Benutzerinteraktionen, E/A-Operationen oder Umgebungseinstellungen. Aus diesen werden die Pfadbeschränkungen für jede erreichbare Anweisung im Programm ermittelt.

Mithilfe von COVA haben wir eine qualitative Studie von mehr als 28.000 Taint-Flows aus einer großen Anzahl reeller kommerzieller Android-Anwendungen durchgeführt. Diese Taint-Flows wurden durch FlowDroid generiert. Wir haben diese Taint-Flows mit den von COVA berechneten Pfadeinschränkungen klassifiziert. Auf diese Weise konnten wir feststellen, wie diese Taint-Flows von Umgebungseinstellungen (z.B. Plattformversionen), Benutzerinteraktionen (z.B. Benutzung von Schaltflächen) und E/A-Operationen (z.B. Lesen und Schreiben von Dateien) abhängen: Etwa 1/3 der Taint-Flows sind falsch-positive Ergebnisse, die verworfen werden sollten. Etwa 4.000 Taint-Flows sind durch die drei von uns definierten Faktoren bedingt. Außerdem können 10% der Taint-Flows zur Laufzeit nur durch bestimmte Benutzerinteraktionen ausgelöst werden. Unsere Ergebnisse deuten darauf hin, dass hybride Ansätze zur dynamischen Validierung statischer Taint-Flows sich auf die Modellierung von Benutzerinteraktionen konzentrieren sollten, aber in begrenztem Umfang auch Konfiguration und E/A berücksichtigen sollten. Basierend auf diesen Erkenntnissen haben wir COVA erweitert, um nicht nur die Pfadbeschränkung einer bestimmten Anweisung zu berechnen, sondern auch konkrete Benutzerinteraktionen zu generieren, die zur Ausführung dieser Anweisung zur Laufzeit erforderlich sind. Experimente mit einer kleinen Auswahl von Anwendungen aus F-Droid zeigen die Machbarkeit dieses Ansatzes zur Generierung gültiger Benutzerinteraktionen, um zufällig ausgewählte Programmanweisungen zu testen. Dieser Ansatz kann zur Verbesserung der Genauigkeit der statischen Taint-Analyse verwendet werden, z.B. nur Taint-Flows, die zur Laufzeit validierte werden können, sollten dem Benutzer gemeldet werden.

5 Geringe Akzeptanz bei Entwicklern

Eine solide, präzise und skalierbare statische Analyse zu erstellen, reicht in der Praxis leider nicht aus. Softwareentwickler sehen sich häufig mit der Aufgabe konfrontiert, eine große Anzahl von Technologien in viel zu kurzer Zeit zu erlernen und anzuwenden. Der sichere

Einsatz dieser Technologien stellt eine große Hürde dar. Oft wird die Sicherheit nicht explizit getestet. Obwohl Sicherheitsanalysen wie die statische Taint-Analyse hier helfen können, werden aktuelle Analysewerkzeuge von den Entwicklern leider nicht gut angenommen. Wie viele neuere Studien zeigen, werden diese Werkzeuge nicht angenommen, wenn sie keine verwertbaren Ergebnisse liefern oder wenn sie diese nicht auf eine Art und Weise darstellen, die für Entwickler verständlich ist und in den Arbeitsablauf von Entwicklern integriert ist [CB16; Do17a; Jo13].

Während statische Taint-Analyse-Werkzeuge Entwicklern dabei helfen können, Sicherheitslücken in ihrem Code zu finden, wurden solche Analysen in interaktiven Entwicklungsumgebungen (IDEs) wie Eclipse, IntelliJ, Android Studio und Visual Studio Code, bisher weniger eingesetzt. Allerdings wären IDEs der ideale Ort für die statische Analyse, und werden von Entwicklern gewünscht. Selbst wenn es eine IDE-Integration gibt, unterstützen Werkzeuge wie DroidSafe [Go15], Cheetah [Do17b] und IBM Security AppScan [IB07] meist nur eine bestimmte IDE, da ein erheblicher technischer Aufwand erforderlich ist, um eine bestimmte Analyse für eine bestimmte Sprache in eine bestimmte IDE zu integrieren. In Anbetracht dieses Aufwands macht es die schiere Vielfalt der gängigen Werkzeuge und potenziell nützlichen Analysen unpraktisch, jede Kombination zu entwickeln. Um eine bessere Akzeptanz dieser Werkzeuge durch Entwickler zu fördern, benötigen Forscher Möglichkeiten, um Werkzeuge einfacher und schneller in IDEs einzubinden. Als vierten Beitrag zu dieser Dissertation haben wir einen allgemeinen Ansatz zur Integration statischer Analysen in IDEs und Editoren entwickelt - MagpieBridge. Um die Verallgemeinerbarkeit von MagpieBridge zu zeigen, haben wir einige Analysen aus dem akademischen Bereich in IDEs integriert—FlowDroid [Ar14], CogniCrypt [Kr17] und Ariadne [Do18], und zwei Analysewerkzeuge aus der Industrie—Facebook Infer [Fa15] und Amazon CodeGuru Reviewer [Se20].

Heutzutage ist statische Taint-Analyse meistens in Static-Application-Security-Testing-Werkzeugen (SAST-Werkzeugen) implementiert. Viele Unternehmen bieten SAST-Werkzeuge als Dienstleistung in der Cloud an. Die aktuellen Lösungen für die Interaktion zwischen Cloud-basierten SAST-Werkzeugen und Entwicklern sind in der Regel webbasiert. Entwickler empfinden es oft als umständlich, zwischen IDE und Webbrowser hin und her zu wechseln, wenn sie die von diesen Werkzeugen erkannten Probleme in ihrem Code beheben wollen. Daher haben wir eine mehrstufige Nutzerstudie mit Software-Ingenieuren bei Amazon Web Services (AWS) durchgeführt, um zu untersuchen, wie die IDE-Unterstützung für ein rein Cloud-basiertes statisches Analysewerkzeug gestaltet sein sollte, um die Erwartungen der Entwickler zu erfüllen. Auf der Grundlage von Interviews mit Entwicklern haben wir einen Prototyp der IDE-Unterstützung für das Cloud-basierte SAST-Werkzeug Amazon CodeGuru Reviewer entwickelt. Wir haben diesen Prototyp mit 32 Software-Ingenieuren bei AWS in einem Usability-Test im Vergleich zur bestehenden webbasierten Lösung evaluiert. Wir fanden heraus, dass die Entwickler mit unserem IDE-Prototyp dreimal häufiger Code-Scans durchführten als mit der webbasierten Lösung. Außerdem konnten sie die von Amazon CodeGuru Reviewer festgestellten Probleme in der IDE schneller beheben. Wir

stellten jedoch fest, dass die Einbindung der Ergebnisse des Werkzeuges in die IDE den Arbeitsablauf der Entwickler nicht uneingeschränkt verbessert hat. Einige Entwickler hatten Schwierigkeiten, die asynchrone Natur der IDE-Lösung zu verstehen. Sie wünschen sich mehr, z.B. Echtzeit-Feedback zum Analysefortschritt, schnelle Validierung von Korrekturen usw.

Die Entwicklung von Werkzeugen zur statischen Taint-Analyse für die Praxis ist eine Herausforderung, die es den Entwicklern von Analysen nicht nur abverlangt, den besten Kompromiss zwischen Genauigkeit, Zuverlässigkeit und Skalierbarkeit zu finden, sondern auch Werkzeuge mit guter Benutzerfreundlichkeit zu entwickeln. In Bezug auf diese Aspekte zeigen wir, dass die statische Taint-Analyse für den Einsatz in der echten Welt verfeinert werden muss und dass sie verbessert werden kann, indem die oben genannten Probleme angegangen werden. Wir hoffen, dass die in dieser Dissertation vorgestellten Benchmarks, Ansätze, ihre Implementierungen und gewonnenen Erkenntnisse den Entwicklern statischer Taint-Analysen dabei helfen können, bessere Werkzeuge zu entwickeln und die Akzeptanz statischer Taint-Analysen bei Softwareentwicklern zu fördern, um sicherere Software zu erstellen.

Literatur

[AL13] Ali, K.; Lhoták, O.: Averroes: Whole-Program Analysis without the Whole Program. In (Castagna, G., Hrsg.): ECOOP 2013 - Object-Oriented Programming - 27th European Conference, Montpellier, France, July 1-5, 2013. Proceedings. Bd. 7920. Lecture Notes in Computer Science, Springer, S. 378–400, 2013, URL: https://doi.org/10.1007/978-3-642-39038-8%5C_16.

[An20] Antoniadis, A.; Filippakis, N.; Krishnan, P.; Ramesh, R.; Allen, N.; Smaragdakis, Y.: Static analysis of Java enterprise applications: frameworks and caches, the elephants in the room. In (Donaldson, A. F.; Torlak, E., Hrsg.): Proceedings of the 41st ACM SIGPLAN International Conference on Programming Language Design and Implementation, PLDI 2020, London, UK, June 15-20, 2020. ACM, S. 794–807, 2020, URL: https://doi.org/10.1145/3385412.3386026.

[Ar14] Arzt, S.; Rasthofer, S.; Fritz, C.; Bodden, E.; Bartel, A.; Klein, J.; Traon, Y. L.; Octeau, D.; McDaniel, P. D.: FlowDroid: precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for Android apps. In: Proceedings of PLDI. ACM, 2014.

[Ar15] Arzt, S.; Rasthofer, S.; Hahn, R.; Bodden, E.: Using targeted symbolic execution for reducing false-positives in dataflow analysis. In (Møller, A.; Naik, M., Hrsg.): Proceedings of the 4th ACM SIGPLAN International Workshop on State Of the Art in Program Analysis, SOAP@PLDI 2015, Portland, OR, USA, June 15 - 17, 2015. ACM, S. 1–6, 2015, URL: https://doi.org/10.1145/2771284.2771285.

[BGC15] Blackshear, S.; Gendreau, A.; Chang, B. E.: Droidel: a general approach to Android framework modeling. In (Møller, A.; Naik, M., Hrsg.): Proceedings of the 4th ACM SIGPLAN International Workshop on State Of the Art in Program Analysis, SOAP@PLDI 2015, Portland, OR, USA, June 15 - 17, 2015. ACM, S. 19–25, 2015, URL: https://doi.org/10.1145/2771284.2771288.

[CB16] Christakis, M.; Bird, C.: What Developers Want and Need from Program Analysis: An Empirical Study. In: Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering. ASE 2016, Association for Computing Machinery, Singapore, Singapore, S. 332–343, 2016, ISBN: 9781450338455, URL: https://doi.org/10.1145/2970276.2970347.

[Do17a] Do, L. N. Q.; Ali, K.; Livshits, B.; Bodden, E.; Smith, J.; Murphy-Hill, E. R.: Cheetah: just-in-time taint analysis for Android apps. In: Proceedings of the 39th International Conference on Software Engineering, ICSE 2017, Buenos Aires, Argentina, May 20-28, 2017 - Companion Volume. S. 39–42, 2017, URL: https://doi.org/10.1109/ICSE-C.2017.20.

[Do17b] Do, L. N. Q.; Ali, K.; Livshits, B.; Bodden, E.; Smith, J.; Murphy-Hill, E. R.: Just-in-time static analysis. In (Bultan, T.; Sen, K., Hrsg.): Proceedings of the 26th ACM SIGSOFT International Symposium on Software Testing and Analysis, Santa Barbara, CA, USA, July 10 - 14, 2017. ACM, S. 307–317, 2017, URL: https://doi.org/10.1145/3092703.3092705.

[Do18] Dolby, J.; Shinnar, A.; Allain, A.; Reinen, J.: Ariadne: analysis for machine learning programs. In (Gottschlich, J.; Cheung, A., Hrsg.): Proceedings of the 2nd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages, MAPL@PLDI 2018, Philadelphia, PA, USA, June 18-22, 2018. ACM, S. 1–10, 2018, URL: https://doi.org/10.1145/3211346.3211349.

[Dr16] DroidBench 3-0, https://github.com/secure-software-engineering/DroidBench/tree/develop, Accessed: 2021-08-03, Sep. 2016.

[Fa15] Facebook: Facebook Infer, https://fbinfer.com, Accessed: 2021-08-03, 2015.

[Go15] Gordon, M. I.; Kim, D.; Perkins, J. H.; Gilham, L.; Nguyen, N.; Rinard, M. C.: Information Flow Analysis of Android Applications in DroidSafe. In: Proceedings of the 22nd NDSS. The Internet Society, 2015.

[IB07] IBM: AppScan, https://www.hcltechsw.com/appscan, Accessed: 2021-08-03, 2007.

[Jo13] Johnson, B.; Song, Y.; Murphy-Hill, E. R.; Bowdidge, R. W.: Why don't software developers use static analysis tools to find bugs? In (Notkin, D.; Cheng, B. H. C.; Pohl, K., Hrsg.): 35th International Conference on Software Engineering, ICSE '13, San Francisco, CA, USA, May 18-26, 2013. IEEE Computer Society, S. 672–681, 2013, URL: https://doi.org/10.1109/ICSE.2013.6606613.

[Ke21] Kersten, J.: LUCA-App – mehr Kosten als Nutzen?, https://www.daserste.de/information/wirtschaft-boerse/plusminus/sendung/sr/sendung-vom-09-06-2021-luca-app-100.html, Accessed: 2021-09-21, 2021.

[Kr17] Krüger, S.; Nadi, S.; Reif, M.; Ali, K.; Mezini, M.; Bodden, E.; Göpfert, F.; Günther, F.; Weinert, C.; Demmler, D. et al.: CogniCrypt: supporting developers in using cryptography. In: Proceedings of the 32nd IEEE/ACM International Conference on Automated Software Engineering. IEEE Press, S. 931–936, 2017.

[LH03] Lhoták, O.; Hendren, L. J.: Scaling Java Points-to Analysis Using SPARK. In (Hedin, G., Hrsg.): Compiler Construction, 12th International Conference, CC 2003, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2003, Warsaw, Poland, April 7-11, 2003, Proceedings. Bd. 2622. Lecture Notes in Computer Science, Springer, S. 153–169, 2003, URL: https://doi.org/10.1007/3-540-36579-6%5C_12.

[Li17] Li, L.; Bissyandé, T. F.; Papadakis, M.; Rasthofer, S.; Bartel, A.; Octeau, D.; Klein, J.; Traon, Y. L.: Static analysis of android apps: A systematic literature review. Inf. Softw. Technol. 88/, S. 67–95, 2017, URL: https://doi.org/10.1016/j.infsof.2017.04.001.

[Lu21] Luo, L.: Improving Real-World Applicability of Static Taint Analysis, Diss., 2021.

[MB08] de Moura, L. M.; Bjørner, N.: Z3: An Efficient SMT Solver. In: Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings. S. 337–340, 2008.

[Me19] Merchan, D.: German politicians' personal information leaked on Twitter in mass cyberattack, https://abcnews.go.com/International/german-politicians-personal-information-leaked-twitter-mass-cyberattack/story?id=60160048, Accessed: 2022-01-31, 2019.

[MK21] Michael D. Shear, N. P.; Krauss, C.: Colonial Pipeline Paid Roughly 5 Million in Ransom to Hackers, https://www.nytimes.com/2021/05/13/us/politics/biden-colonial-pipeline-ransomware.html, Accessed: 2021-08-03, 2021.

[Re21] Reuter, M.: Schon wieder desaströse Sicherheitslücke in Luca App, https://netzpolitik.org/2021/it-sicherheit-schon-wieder-desastroese-sicherheitsluecke-in-luca-app, Accessed: 2021-09-21, 2021.

[Se20] Services, A. W.: Amazon CodeGuru Reviewer, https://aws.amazon.com/codeguru, Accessed: 2021-08-03, 2020.

[Sr11] Sridharan, M.; Artzi, S.; Pistoia, M.; Guarnieri, S.; Tripp, O.; Berg, R.: F4F: taint analysis of framework-based web applications. In (Lopes, C. V.; Fisher, K., Hrsg.): Proceedings of the 26th Annual ACM SIGPLAN Conference on Object-Oriented Programming, Systems, Languages, and Applications, OOPSLA 2011, part of SPLASH 2011, Portland, OR, USA, October 22 - 27, 2011. ACM, S. 1053–1068, 2011, URL: https://doi.org/10.1145/2048066.2048145.

[St20] Statista: Annual number of data breaches and exposed records in the United States from 2005 to 2020, https://www.statista.com/statistics/273550/data-breaches-recorded-in-the-united-states-by-number-of-breaches-and-records-exposed, Accessed: 2021-08-03, 2020.

[Su00] Sundaresan, V.; Hendren, L. J.; Razafimahefa, C.; Vallée-Rai, R.; Lam, P.; Gagnon, E.; Godin, C.: Practical virtual method call resolution for Java. In (Rosson, M. B.; Lea, D., Hrsg.): Proceedings of the 2000 ACM SIGPLAN Conference on Object-Oriented Programming Systems, Languages & Applications (OOPSLA 2000), Minneapolis, Minnesota, USA, October 15-19, 2000. ACM, S. 264–280, 2000, URL: https://doi.org/10.1145/353171.353189.

[We14] Wei, F.; Roy, S.; Ou, X.; Robby: Amandroid: A Precise and General Inter-component Data Flow Analysis Framework for Security Vetting of Android Apps. In: Proceedings of CCS. ACM, 2014.

[WR13] Wei, S.; Ryder, B. G.: Practical blended taint analysis for JavaScript. In (Pezzè, M.; Harman, M., Hrsg.): International Symposium on Software Testing and Analysis, ISSTA '13, Lugano, Switzerland, July 15-20, 2013. ACM, S. 336–346, 2013, URL: https://doi.org/10.1145/2483760.2483788.

Linghui Luo ist derzeit als angewandte Wissenschaftlerin bei Amazon Web Services in Berlin tätig. Nach ihrem Schulabschluss in China kam Luo nach Deutschland und studierte sie Informatik an der Universität Paderborn. Sie schloss jeweils in den Jahren 2014 und 2017 ihren Bachelor und Master an der Universität Paderborn ab. Danach arbeitete sie bis Ende 2021 als wissenschaftliche Mitarbeiterin in der Fachgruppe von Prof. Eric Bodden an der Universität Paderborn. 2021 promovierte sie mit dem Thema "Improving Real-World Applicability of Static Taint Analysis". Mit ihrer Forschung gewann sie die Silbermedaille in der ACM Student Research Competition auf der ESEC/FSE-Konferenz 2021. Während ihrer Promotion absolvierte sie zwei Praktika in der Industrie: IBM Thomas J. Watson Research Center im Jahr 2019 und Amazon Web Services im Jahr 2020. In ihrer Forschung beschäftigt sich Linghui Luo mit Programmanalyse mit den Schwerpunkten Softwaresicherheit, empirische Softwareentwicklung und nutzbare Sicherheit.

Berechnung effizienter Datenzusammenfassungen[1]

Sebastian Mair[2]

Abstract:

Das Extrahieren sinnvoller Repräsentationen von Daten ist ein grundlegendes Problem im maschinellen Lernen und kann aus zwei unterschiedlichen Perspektiven betrachtet werden: (i) im Bezug auf die Anzahl der Datenpunkte und (ii) hinsichtlich der Repräsentation eines jeden einzelnen Datenpunktes in Bezug auf seine Dimensionen. Diese Arbeit beschäftigt sich mit diesen Perspektiven zur Datenrepräsentation und leistet dazu verschiedene Beiträge. Der erste Teil behandelt die Berechnung repräsentativer Teilmengen für die Archetypenanalyse und die Problemstellung der optimalen Versuchsplanung. Dafür motivieren und untersuchen wir die Brauchbarkeit der Punkte am Rand der Daten als neuartige repräsentative Teilmenge. Basierend auf dem Coreset-Prinzip leiten wir eine weitere repräsentative Teilmenge für die Archetypenanalyse her, welche zusätzliche theoretische Garantien bietet. Der zweite Teil der Arbeit handelt von effizienten Datenrepräsentationen für Dichteschätzungsprobleme. Wir analysieren raum-zeitliche Probleme, die z.B. in der Analyse von Mannschaftssportarten auftreten, und zeigen, wie sich statistische Bewegungsmodelle anhand von Trajektoriendaten lernen lassen. Darüber hinaus untersuchen wir Probleme hinsichtlich der Interpolation von Daten mittels generativer Modelle.

1 Einführung

Im maschinellen Lernen geht es hauptsächlich um das Auffinden von Strukturen und Mustern in Daten, sowie um das Finden von funktionalen Zusammenhängen zwischen Eingabe- und Ausgabedaten. Oft werden dabei Objekte aus der realen Welt als Vektoren repräsentiert, wobei die Dimensionen, auch Attribute genannt, verschiedene Merkmale der Objekte beschreiben. Eine Menge dieser Vektoren, auch Datenpunkte genannt, bildet dann den Datensatz. Aus diesem lernt ein Verfahren des maschinellen Lernens dann Muster bzw. Zusammenhänge. Dieser Datensatz wird üblicherweise in Form einer Matrix dargestellt. In dieser Datenmatrix liegen die Datenpunkte aufeinander gestapelt.

Das am weitesten verbreitetste Lernszenario ist das des überwachten Lernens. Hierbei besitzt jeder einzelne Datenpunkt zusätzlich einen Zielwert. Das Ziel ist das Lernen einer Funktion, welche neuen Datenpunkten automatisch passende Zielwerte zuordnet. Man spricht von einem Regressionsproblem, wenn der Zielwert eine reelle Zahl ist. Ist die Menge an möglichen Zielwerten jedoch diskret, betrachtet man ein Klassifikationsproblem. Im Allgemeinen vereinfacht das Vorhandenseins eines Zielwertes sowohl das Lernen, als auch das Evaluieren eines Modells. Jedoch ist ein solcher Zielwert nicht immer gegeben, so z.B.

[1] Englischer Titel der Dissertation: „Computing Efficient Data Summaries“

[2] Universität Uppsala, Schweden, sebastian.mair@it.uu.se

beim unüberwachten Lernen. Hierbei ist das Ziel das Finden von Strukturen und Mustern in den Daten. Typische Beispiele von unüberwachten Lernverfahren sind Anomalieerkennung, Dimensionalitätsreduktion und das Lernen von Wahrscheinlichkeitsverteilungen.

Ein entscheidender Faktor für die Leistungsfähigkeit eines Modells ist mitunter die Darstellung der Daten. Diese Repräsentation, im Sinne des Datensatzes bzw. der Datenmatrix, ist jedoch oft suboptimal. So kann der Datensatz beispielsweise viel zu groß für die vorhandene Infrastruktur sein, da er z.B. zu viele unnötige Redundanzen beinhaltet. Ebenfalls können vorhandene Attribute wenig oder gar keine Vorhersagekraft besitzen. In beiden Fällen ist eine Repräsentationsänderung vorteilhaft. Diese Änderung kann entweder als Teil der Datenvorverarbeitung erfolgen, oder ein Teil des Lernalgorithmus sein. Im Allgemeinen gibt es zwei Sichtweisen auf die Daten, bei denen eine Änderung der Repräsentation erfolgen kann.

2 Die zwei Sichtweisen des Repräsentationslernens

Beim Repräsentationslernen geht es darum, eine bessere Repräsentation der Daten zu finden. So kann z.B. die Datenmatrix für ein bestimmtes Lernszenario spezifisch angepasst werden. Diese Änderung der Repräsentation kann auf zwei unterschiedliche Arten betrachtet werden. Bei der einen Sichtweise geht es um die Anzahl der Datenpunkte und somit um die Stichprobengröße des Datensatzes während die andere von einer Änderung der Repräsentation in Bezug auf die Attribute handelt.

Die Repräsentation des Datensatzes im Bezug auf die Anzahl der Datenpunkte

In vielen Problemstellungen sind Daten in großen Mengen vorhanden und beinhalten viele redundante Informationen. Dabei kann die Größe des Datensatzes im Sinne der Anzahl der Datenpunkte zu groß sein, um bestimmte Lernverfahren effizient anzuwenden. Abgesehen davon kann es auch sein, dass Speicher- und Berechnungsbeschränkungen uns dazu zwingen, eine Approximation des Lernverfahrens zu verwenden. Üblicherweise werden dann mit erheblichem Mehraufwand aufwändige Approximationen der Lernverfahren verwendet, um diese auf große Datenmengen anwenden zu können. Alternativ kann statt des Modells auch der zugrundeliegende Datensatz approximiert werden, womit wir uns in dieser Arbeit näher beschäftigen.

Wir nennen eine Teilmenge des Datensatzes *repräsentative Teilmenge* oder *Datenzusammenfassung*, wenn diese die Hauptcharakteristiken der Ursprungsdaten gut approximiert. Trivialerweise kann man unabhängig und gleich-verteilte Datenpunkte nach dem Zufallsprinzip aus dem Datensatz auswählen. Der Vorteil dieses Ansatzes liegt in seiner Einfachheit. Darüber hinaus liefert der Ansatz einen unverzerrten Schätzer der zu optimierenden Zielfunktion. Im folgenden werden wir jedoch zeigen, dass nur mit minimalem Mehraufwand repräsentativere Teilmengen generiert werden können, welche bei gleicher Stichprobengröße, erheblich bessere Approximationen liefern.

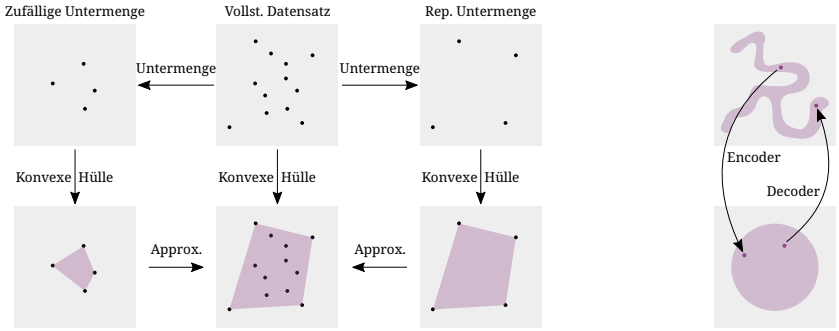


Abb. 1: Links: Eine repräsentative Untermenge für das Lernen einer konvexen Hülle. Rechts: Eine Datentransformation wie sie z.B. bei einem VAE vorkommt.

Abbildung 1 (links) zeigt Beispiele von repräsentativen Untermengen für die Berechnung einer konvexen Hülle. Eine Teilmenge bestehend aus unabhängig und gleich-verteilten Datenpunkten deckt einen kleineren Bereich ab als die gewünschte konvexe Hülle des vollständigen Datensatzes. Dies ist darauf zurückzuführen, dass die wichtigsten Punkte für dieses Problem am Rand der Daten liegen. Im Vergleich dazu enthält die repräsentative Teilmenge auf der rechten Seite der Abbildung mehr relevante Punkte und bietet somit eine bessere Zusammenfassung der vollständigen Daten. Infolgedessen liefert sie eine viel bessere Annäherung bei gleicher Größe der verwendeten Teilmenge.

Die Repräsentation eines jeden einzelnen Datenpunktes

Üblicherweise wird Repräsentationslernen als eine Änderung der Repräsentation in Bezug auf die Dimensionen der Daten verstanden. So beinhalten z.B. nicht alle Dimensionen sinnvolle Informationen für die Lernaufgabe, oder die wesentlichen Informationen sind implizit in einem Unterraum eingebettet. Häufig kann eine Transformation der Datenrepräsentation das Lernverfahren an sich bzw. darauf aufbauende Verfahren vereinfachen.

Ein Beispiel für nicht-lineare Merkmalstransformationen sind Autoencoder, welche aus zwei neuronalen Netzen bestehen. Das erste Netz, der Encoder, bettet die Eingabedaten in einen sogenannten *latenten Raum* ein. Die Eingabedaten können dann durch die Repräsentationen im latenten Raum durch das zweite Netz, Decoder genannt, rekonstruiert werden. Die latente Repräsentation ist oft von geringerer Dimensionalität und zwingt den Autoencoder die wesentlichen Charakteristiken und Eigenschaften der Daten zu extrahieren.

Ein Autoencoder kann auch zur Generierung neuer Daten, welche ähnlich zu den Trainigsdaten sind, verwendet werden. Hierfür ist es nötig, dem latenten Raum eine bestimmte Struktur aufzuzwingen. Dies realisiert beispielsweise ein Variational Autoencoder (VAE), welcher üblicherweise eine Normalverteilung im latenten Raum erzwingt. Wir erhalten somit ein generatives Modell, in welchem wir Realisierungen der Normalverteilung mittels

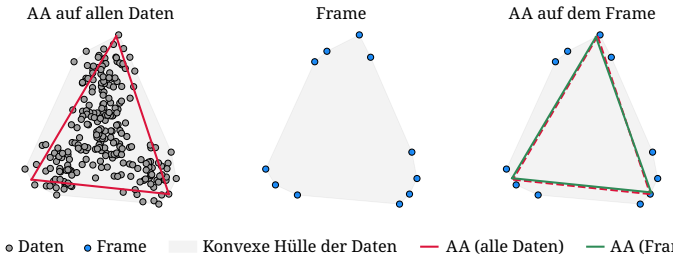


Abb. 2: Eine Archetypenanalyse mit drei Faktoren berechnet auf allen Daten (links), der Frame der Daten (mitte) und die Archetypenanalyse berechnet auf dem Frame (rechts).

des Decoders in neue, synthetische Datenpunkte transformieren können. Ein Beispiel ist in Abbildung 1 (rechts) dargestellt. Eine komplexe Datenverteilung wird in eine einfachere Verteilung umgewandelt. Dies ist nicht nur für die generative Modellierung, sondern auch für die Dichteschätzung vorteilhaft.

3 Beiträge dieser Arbeit

Die kummulative Dissertation [Ma22] beinhaltet sechs Einzelpublikationen und leistet verschiedene Beiträge im Bereich des unüberwachten Repräsentationslernens. Dabei werden insbesondere die zwei zuvor eingeführten Sichtweisen behandelt und pro Sichtweise eine Forschungsfrage abgeleitet. Der erste Teil der Arbeit beschäftigt sich mit der Frage, wie man effizient repräsentative Teilmengen für Lernaufgaben berechnen kann, welche es ermöglichen, dasselbe aus weniger Daten auf effizientere Weise zu lernen. Im zweiten Teil stellt sich die Frage, wie man Datenrepräsentationen erhält, welche die Anwendung spezifischer Operationen wie Dichteschätzung und Interpolation erleichtern. Im Folgenden wird pro Forschungsfrage eine Auswahl an Beiträgen der Dissertation vorgestellt.

3.1 Der Frame als repräsentative Untermenge

Wir untersuchen zunächst die Idee den Rand der Daten, im folgenden *Frame* genannt, als repräsentative Untermenge zu verwenden. Die Hypothese ist, dass für bestimmte lineare Lernverfahren die Datenpunkte am Rand bereits genügend Information für das Lernproblem tragen, sodass eine Restriktion auf den Rand zufriedenstellende Ergebnisse liefert. Dazu betrachten wir zunächst das Problem der Archetypenanalyse (AA) [CB94]. Das Ziel ist eine Matrixfaktorisierung der Datenmatrix in eine Gewichts- und Faktormatrix. Die Idee ist jeden Datenpunkt als Konvexkombination der Faktoren, hier *Archetypen* genannt, darzustellen. Hierbei sind die Archetypen selbst Konvexkombinationen aus Datenpunkten. Die Gewichte können nun als Wahrscheinlichkeitsverteilung interpretiert werden, was eine zusätzliche Interpretierbarkeit der Faktorisierung ermöglicht. Es lässt sich zeigen, dass für diese Matrixfaktorisierung, die Archetypen immer am Rand der konvexen Hülle liegen [CB94].

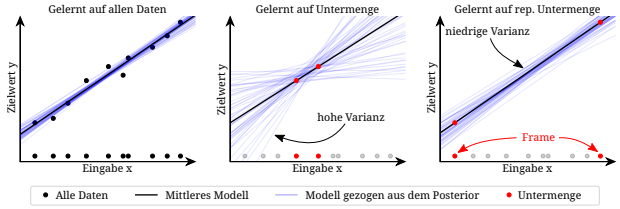


Abb. 3: Ein Beispiel für eine optimale Versuchsplanung in einer Dimension.

In [MBB17] zeigen wir, dass die Restriktion der Berechnung der Archetypenanalyse auf den Frame nahezu die selben Resultate liefert, wie die Berechnung der Faktorisierung auf allen Daten. Siehe dazu Abbildung 2. Zusätzlich wird durch die Restriktion die Berechnung der Archetypenanalyse erheblich beschleunigt.

Der Frame kann mit gängigen Algorithmen zur Bestimmung von konvexen Hüllen berechnet werden. Die Mehrheit dieser Algorithmen ist jedoch für nur zwei- bzw. dreidimensionale Probleme ausgelegt. Verfahren für höherdimensionale Datensätze sind ineffizient, da nicht nur die Randpunkte, sondern auch die überflüssigen Facetten berechnet werden. Als Abhilfe stellen wir in [MBB17] ein neues Verfahren zur Berechnung des Frames in Räumen mit höherer Dimensionalität vor. Unser Ansatz basiert auf quadratischer Programmierung und zeigt eine interessante Verbindung zum Optimierungsalgorithmus NNLS [LH95], welcher zur Lösung von Regressionsproblemen mit nicht-negativen Lösungsvektoren eingesetzt wird.

Ein zweites Lernproblem, welches von der Verwendung des Frames als repräsentative Untermenge profitiert, ist die optimale Versuchsplanung [Fe72]. Hierbei ist eine Menge von Datenpunkten gegeben, welche Experimente beschreiben. Zusätzlich wird ein linearer Zusammenhang zwischen einem parametrisierten Experiment und dem Resultat des durchgeführten Experimentes angenommen. Das Ziel ist es, bei vorgegebener Versuchsanzahl, jene Experimente auszuwählen und diese tatsächlich durchzuführen, sodass mit den Resultaten eine lineare Regression die best möglichste Vorhersage der nicht ausgeführten Experimente liefert.

Wir zeigen in [Ma18], dass auch hier die Einschränkung der Berechnung auf den Rand der Daten (Frame) den Berechnungsaufwand, bei nahezu gleichbleibender Qualität der Vorhersagequalität, deutlich verringert. Abbildung 3 liefert eine Intuition weshalb der Frame eine sinnvolle repräsentative Untermenge liefert. Hier wird die Varianz der Parameterschätzung durch die Wahl von Randpunkten minimiert. Im selben Forschungspapier zeigen wir Verbindungen des Frames zu geometrischen Interpretationen verschiedener Optimalitätskriterien, welche in der optimalen Versuchsplanung vorwiegend verwendet werden. Darüber hinaus, beschäftigen wir uns mit der nicht-linearen Erweiterung der Problemstellung. Die Regression kann durch Verwendung von Kern-Funktionen [SS02] auch nicht-lineare Zusammenhänge abbilden. Diesbezüglich adaptieren wir die Berechnung des Frames und zeigen wie dieser auch in kern-induzierten Merkmalsräumen bestimmt werden kann. Ebenfalls analysieren wir die erwartete Anzahl an Randpunkten für gängige Kerne theoretisch und diskutieren eine Verbindung der Frame-Berechnung mit dem LASSO-Verfahren [Ti96] zur Merkmalsselektion.

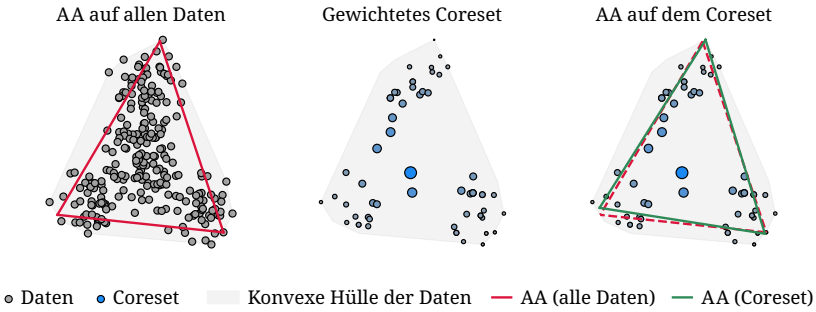


Abb. 4: Eine Archetypenanalyse mit drei Faktoren berechnet auf allen Daten (links), ein gewichtetes Coreset (mitte) und die Archetypenanalyse berechnet auf dem Coreset (rechts).

3.2 Coresets für die Archetypenanalyse

Obwohl der Frame als repräsentative Teilmenge den Rechenaufwand der Archetypenanalyse reduziert und dennoch eine kompetitive Lösung geliefert hat, besitzt diese Teilmenge einige Nachteile. Zunächst ist die Frame-Berechnung für hochdimensionale Datensätze problematisch, da die Berechnung polynomiell in der Dimensionalität der Daten skaliert. Zweitens ist die Größe der repräsentativen Teilmenge eine inhärente Eigenschaft des Datensatzes und kann somit nicht frei gewählt werden. Schließlich gibt es keine theoretische Fehlerabschätzung, sowie eine Garantie, dass die Verwendung des Frames tatsächlich eine kompetitive Lösung liefert.

Diesem Problem widmen wir uns speziell für die Archetypenanalyse in [MB19] unter Verwendung des Coreset-Frameworks. Hierbei bezeichnet man eine (gewichtete) Untermenge eines Datensatzes als *Coreset*, wenn ein Modell welches auf der Untermenge trainiert wurde, nachweislich mit dem Modell welches auf allen Daten trainiert wurde, konkurrieren kann. Die Idee ist, dass ein Datenpunkt stellvertretend für mehrere Datenpunkte steht und dementsprechend ein Gewicht zugeordnet wird. Ebenfalls sollen wichtige Bereiche des Eingaberaums mit mehr Punkten abgedeckt werden, als unwichtige. Üblicherweise sind Coresets effizient in linearer Zeit berechenbar und basieren auf probabilistischen Verfahren, welche eine Stichprobenziehung nach Wichtigkeit ermöglichen.

Die Mitte von Abbildung 4 zeigt ein Beispiel eines Coresets mit gewichteten Datenpunkten. Für das Problem der Archetypenanalyse haben wir eine Wahrscheinlichkeitsverteilung vorgeschlagen, welche Datenpunkte am Rand mit höherer Wahrscheinlichkeit, jedoch mit einem kleineren Gewicht auswählt. Im Gegensatz dazu werden Datenpunkte im Zentrum des Datensatzes seltener ausgewählt, bekommen aber ein größeres Gewicht. Für die Berechnung dieser Punkte muss der Algorithmus lediglich zweimal über den Datensatz iterieren. Somit ist die Berechnung der repräsentativen Untermenge sehr effizient. Des Weiteren haben wir eine Abschätzung hergeleitet, welche die Größe des Coresets in Verbindung mit dem Fehler der Approximation und der Wahrscheinlichkeit des Einhaltes dieser Abschätzung setzt.

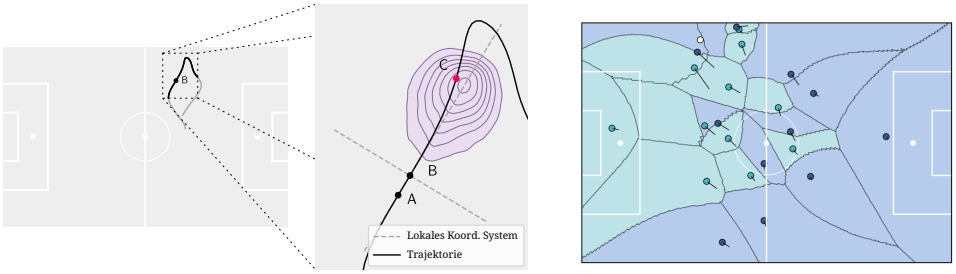


Abb. 5: Ein statistisches Bewegungsmodell (links) und von den Spielern kontrollierte Zonen (rechts).

Neben zahlreichen Experimenten zur empirischen Verifikation der Güte der repräsentativen Teilmenge zeigen wir ebenfalls, dass sich die Zielfunktion der Archetypenanalyse mit der Zielfunktion des k-Means Verfahren zur Clusteranalyse [Ll82] abschätzen lässt. Dies zeigt nicht nur eine interessante Verbindung beider unüberwachten Lernverfahren, sondern ermöglicht auch die Verwendung von Coresets, welche explizit für das k-Means Verfahren erstellt wurden, für die Archetypenanalyse.

3.3 Statistische Bewegungsmodelle

Der zweite Teil der Dissertation behandelt eine Repräsentationsänderung im Bezug auf die Dimensionalität der Daten mit dem Ziel, spezifische Operationen, wie z.B. die Dichteschätzung oder die Interpolation in generativen Modellen, zu verbessern.

Diesbezüglich befassen wir uns nun mit der Dichteschätzung auf Trajektorien in einem raum-zeitlichen Kontext. Speziell geht es um die Erstellung von statistischen Bewegungsmodellen von Objekten, also um die Quantifizierung der Wahrscheinlichkeit für die nächste Position mit einem gewissen Zeithorizont, gegeben verschiedener Kontextinformationen wie z.B. die Bewegungsrichtung, die momentane Geschwindigkeit sowie die Position anderer Objekte. Solche Probleme sind beispielsweise in der Koordination von Spielern in Mannschaftssportarten, beim Studieren von Migrationsmustern oder bei Tierwanderungen anzutreffen. In dieser Arbeit beschäftigen wir uns speziell mit Fußballdaten.

Die linke Seite von Abbildung 5 zeigt ein Beispiel für ein statistisches Bewegungsmodell. Ein Spieler läuft von Position A zu Position B. Das Bewegungsmodell liefert nun, gegeben der Bewegungsrichtung und Geschwindigkeit eines Spielers, eine Verteilung für die Position, welche der Spieler in einer vorgegebenen Zeit erreichen wird. Die tatsächliche Position C ist rot hervorgehoben. Traditionelle Modelle basierend auf physikalischen Annahmen, haben oft unrealistische Nebeneffekte und sind selten personalisiert. In [BLM19] haben wir das in Abbildung 5 (links) gezeigte Bewegungsmodell vorgeschlagen. Das Modell basiert auf Kerndichteschätzern und kann aus Bewegungsdaten personalisiert gelernt werden. Hierbei werden Trajektoriendaten in ein lokales Koordinatensystem überführt, welches die Dichteschätzung drastisch vereinfacht.

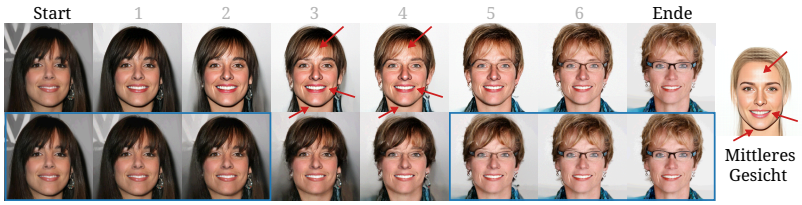


Abb. 6: Zwei Interpolationspfade eines generativen Modells am Beispiel von Gesichtern.

Kerndichteschätzer als nicht-parametrische Methoden haben den Vorteil, dass die Güte der Vorhersage mit jedem zusätzlichen Datenpunkt zunimmt. Der Nachteil ist jedoch, dass der Berechnungsaufwand einer Vorhersage ebenfalls ansteigt. Deshalb haben wir in [Fa21a] das lokale Koordinatensystem mittels invertierbarer neuronaler Netze in geeignete Repräsentationen überführt. Dies hat mehrere Vorteile: die Komplexität einer Vorhersage ist statisch und nicht länger abhängig von der Größe des Datensatzes, die Vorhersagequalität steigt, und es sind komplexere Kontextinformationen möglich. Während das Modell basierend auf Kerndichteschätzern nur die Bewegungsrichtung und Geschwindigkeit berücksichtigt hat, kann das neue Modell in [Fa21a] auch auf die Positionen der anderen Spieler konditioniert werden. Empirische Experimente auf Fußballdaten haben gezeigt, dass durch die Anreicherung an Kontextinformation die Vorhersagegüte drastisch steigt.

In [BLM19] haben wir ebenfalls gezeigt, wie sich statistische Bewegungsmodelle zur Berechnung von kontrollierten Zonen auf dem Spielfeld verwenden lassen. Hierbei ist eine, von einem Spieler kontrollierte Zone, jener Bereich, welcher von diesem Spieler am wahrscheinlichsten zuerst erreicht wird. Ein Beispiel ist in Abbildung 5 auf der rechten Seite abgebildet. Jeder Spieler hat einen eigenen kontrollierten Bereich. Die Striche geben an, aus welcher Richtung der Spieler kommt. Solche kontrollierten Zonen ermöglichen eine neuartige, datengetriebene Analyse von Spielen.

3.4 Interpolieren in generativen Modellen

Im Beitrag [Fa21b] beschäftigen wir uns mit generativen Modellen basierend auf tiefen neuronalen Netzen. Wie in Abbildung 1 (rechts) angedeutet, ist die Idee Daten, welche einer komplizierten Verteilung folgen, in eine andere Repräsentation zu überführen, welche vorgegebenen Strukturen folgt. Abbildung 6 zeigt zwei Interpolationspfade eines generativen Modells, welches eine Standardnormalverteilung im latenten Raum annimmt. Dabei sind die Gesichter am Rand normale Datenpunkte während die sechs Gesichter dazwischen synthetische Bilder entlang der Interpolationspfade sind. Rechts der Interpolation ist der dekodierte Erwartungswert des generativen Modells abgebildet. Es zeigt somit das „mittlere Gesicht“. Der obere Interpolationspfad resultiert aus einer einfachen linearen Interploation. Wie mit den roten Pfeilen hervorgehoben, nehmen die Gesichter in der Mitte Eigenschaften des Erwartungswertes an. So ist z.B. eine glänzende Stirn in keinem der Ursprungsbilder,

jedoch im „mittlere Gesicht“ vorhanden. Der Grund hierfür ist, dass in hochdimensionalen Räumen standardnormalverteilte Datenrepräsentationen einen gewissen Abstand zum Zentrum des Koordinatensystems haben. Interpolanten einer linearen Interpolation haben jedoch einen wesentlich geringeren Abstand und sind somit dem Zentrum näher. Dieses Zentrum ist allerdings der Erwartungswert der Standardnormalverteilung. Im unteren Interpolationspfad in Abbildung 6 wird zusätzlich der Abstand der Interpolanten zum Zentrum interpoliert. Damit behält der Interpolationspfad den erwarteten Abstand zum Zentrum und das vorherige Problem tritt nicht länger auf. Allerdings gibt es ein neues Problem. Da sich die Interpolationsgeschwindigkeit ändert, gibt es eine Verzerrung hin zu den Randbildern (in blau hervorgehoben) und die Interpolation wirkt ungleichmäßig.

Diesem Problem widmen wir uns in [Fa21b]. Wir schlagen für den latenten Raum die Verwendung einer Einheitssphäre vor, in welchem die Interpolationen als geodätische Wege effizient berechnet werden können. Dabei verwenden wir eine stereografische Projektion um die Daten aus einem euklidischen Raum in die Einheitssphäre zu transformieren. Auf der Einheitssphäre verwenden wir eine von Mises-Fisher Verteilung als Pendant einer Normalverteilung. Zahlreiche quantitative und qualitative Experimente zeigen, dass die somit gewonnenen Interpolationspfade natürlicher Wirken und nicht von den in Abbildung 6 gezeigten Problemen betroffen sind. Dabei wird durch die Verbesserung der Interpolationsqualität die Leistungsfähigkeit des generativen Modells nicht beeinflusst. Die Qualität der Erstellung beliebiger neuer Datenpunkte bleibt erhalten.

4 Fazit und Ausblick

Diese Dissertation befasst sich mit dem unüberwachten Lernen von effizienten Datenrepräsentationen. Dabei wurde das Problem aus zwei orthogonalen Sichtweisen betrachtet: (i) in Bezug auf den Stichprobenumfang und (ii) hinsichtlich der Dimensionen eines jeden Datenpunktes. Zu beiden Sichtweisen wurden mehrere Beiträge geleistet und neue Ansätze, Methoden, Verbindungen zwischen unterschiedlichen Lernproblemen, sowie Probleme und deren Lösungen aufgezeigt. Die einzelnen Beiträge wurden durch Experimente und theoretischen Analysen untermauert.

Zukünftige Arbeiten widmen sich einer Kombination beider Sichtweisen, anstatt diese separat voneinander zu betrachten. So sind in Bezug auf die stetig wachsende Datenbasis effiziente Repräsentationen ebendieser notwendig, die sowohl in Anzahl, als auch in Dimensionalität kompakter sind, dabei aber nicht an Aussagekraft verlieren.

Literatur

[BLM19] Brefeld, U.; Lasek, J.; Mair, S.: Probabilistic movement models and zones of control. Machine Learning 108/1, S. 127–147, 2019.

[CB94] Cutler, A.; Breiman, L.: Archetypal analysis. Technometrics 36/4, S. 338–347, 1994.

[Fa21a] Fadel, S. G.; Mair, S.; da Silva Torres, R.; Brefeld, U.: Contextual movement models based on normalizing flows. AStA Advances in Statistical Analysis/, S. 1–22, 2021.

[Fa21b] Fadel, S. G.; Mair, S.; da Silva Torres, R.; Brefeld, U.: Principled Interpolation in Normalizing Flows. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, S. 116–131, 2021.

[Fe72] Fedorov, V. V.: Theory of optimal experiments. Elsevier, 1972.

[LH95] Lawson, C. L.; Hanson, R. J.: Solving least squares problems. SIAM, 1995.

[Ll82] Lloyd, S.: Least squares quantization in PCM. IEEE transactions on information theory 28/2, S. 129–137, 1982.

[Ma18] Mair, S.; Rudolph, Y.; Closius, V.; Brefeld, U.: Frame-Based Optimal Design. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, Cham, S. 447–463, 2018.

[Ma22] Mair, S.: Computing Efficient Data Summaries, Diss., Leuphana Universität Lüneburg, 2022.

[MB19] Mair, S.; Brefeld, U.: Coresets for Archetypal Analysis. In: Advances in Neural Information Processing Systems. S. 7247–7255, 2019.

[MBB17] Mair, S.; Boubekki, A.; Brefeld, U.: Frame-based data factorizations. In: International Conference on Machine Learning. PMLR, S. 2305–2313, 2017.

[SS02] Schölkopf, B.; Smola, A. J.: Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, 2002.

[Ti96] Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological)/, S. 267–288, 1996.

Sebastian Mair, geboren 1988 in Dillingen an der Donau, studierte zunächst Informatik an der Hochschule Darmstadt, bevor er ein Masterstudium in Informatik sowie ein Bachelorstudium in Mathematik an der Technischen Universität Darmstadt absolvierte. Im Anschluss promovierte er im maschinellen Lernen unter der Betreuung von Ulf Brefeld an der Leuphana Universität in Lüneburg, an welcher er auch als Wissenschaftlicher Mitarbeiter beschäftigt war. Seine Promotion schloss er im September 2021 mit *summa cum laude* ab. Im Dezember 2021 startete er als Postdoktorand an der Universität Uppsala in Schweden. Seine Forschungsinteressen sind unüberwachtes Lernen, effiziente Datenrepräsentationen, generative Modellierung und statistisches maschinelles Lernen.

Steffen Hölldobler et al. (Hrsg.): Ausgezeichnete Informatikdissertationen 2021,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2022 201

Verifikation von Markov Entscheidungsprozessen in diskreter Zeit[1]

Tobias Meggendorfer[2]

Abstract: Diese Arbeit beschäftigt sich mit der Verifikation von probabilistischen Systemen in diskreter Zeit, insbesondere *Markov Entscheidungsprozessen* (MDP). Sie beinhaltet sowohl theoretische als auch praktische Fortschritte, aufgeteilt in vier Bereiche. Zuerst wird eine jahrzehntealte offene Frage bezüglich *mean payoff* Problemen gelöst und, basierend auf der Antwort, neue Algorithmen präsentiert, die als erste solche Probleme effizient berechnen. Dann wird eine effiziente und flexible Implementierung von *LTL-zu-Automaten* Übersetzungsalgorithmen gezeigt, die unter Anderem im Kontext der probabilistischen Verifikation von LTL Formeln eine zentrale Rolle spielt. Als Drittes wird das fundamental neue Konzept des *Kerns* eines MDP präsentiert, eine flexible Grundlage um approximative Berechnungen auf MDP zu beschleunigen. Zuletzt wird ein wichtiger Schritt in Richtung risikobewusster Analyse von probabilistischen Systemen diskutiert.

1 Einleitung

Das zentrale Objekt dieser Arbeit sind probabilistische Systeme, insbesondere Markov Entscheidungsprozesse. Die unterlegenden Fragestellungen entspringen zahlreichen Anwendungsbereichen wie zum Beispiel randomisierten Kommunikationsprotokolle, stochastischen verteilten Systemen, biologischen und chemischen Prozessen, künstlicher Intelligenz, Spracherkennung, Bewegungsplanung, und viele mehr. Eine Übersicht über konkrete Anwendungen ist bspw. in [Wh93] zu finden. Die kennzeichnende Eigenschaft solcher Systeme ist die Vereinigung von sowohl stochastischen als auch nicht-deterministischen Elementen. Erstere beschreiben den Einfluss der Umgebung, also Ereignisse die außerhalb der Kontrolle des analysierten Entscheiders liegen, jedoch indifferent und nicht antagonistisch sind. Zweitere werden verwendet um Entscheidungsmöglichkeiten zu modellieren. Zum Beispiel kann ein Roboter sich entscheiden in die eine oder andere Richtung zu steuern, der genaue Effekt der Bewegung ist jedoch potentiell unklar, beispielsweise aufgrund von glattem Untergrund oder Ungenauigkeiten in den Aktuatoren.

Markov Entscheidungsprozesse (MDP) [Pu94] sind ein klassisches und elegantes mathematisches Werkzeug zur Modellierung solcher Situationen. Im wesentlichen haben diese drei Bestandteile: Zustände, Aktionen und probabilistische Übergänge bzw. Transitionen zwischen den Zuständen. Zustände sind eine vollständige Momentaufnahme des Systems, beispielsweise die Position, Ausrichtung und Geschwindigkeit eines Roboters.

[1] Englischer Titel der Dissertation: „Verification of Discrete-Time Markov Decision Processes“

[2] IST Austria, Am Campus 1, 3400 Klosterneuburg, Austria. tobias@meggendorfer.de

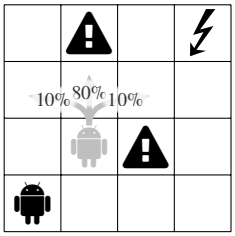


Abb. 1: Das „Grid World“ Modell. Ein ferngesteuerter Roboter befindet sich in der linken unteren Ecke, seine Ladestation oben rechts, und zwei Gefahrenstellen dazwischen. Die möglichen Ergebnisse der „Norden“ Aktion an Position $(2, 2)$ sind in Grau skizziert.

In jedem Zustand steht eine Menge an Aktionen zur Verfügung, z.B. „Beschleunigen“, „Bremsen“ oder „Lenken“. Diese Aktionen haben einen (potentiell stochastischen) Einfluss auf den Zustand des Systems. Im Kontext von MDP wird angenommen, dass die Wahrscheinlichkeitsverteilung dieses Einflusses a-priori bekannt ist.

Ein minimalistisches Beispiel für MDP ist das sogenannte „Grid World“ Modell. Hier wird ein Roboter auf einem Gitter (beispielsweise von Größe 4x4) modelliert. Dieser Roboter kann ferngesteuert werden um verschiedene Aufgaben zu erfüllen. Im Beispiel befindet sich noch eine Ladestation sowie Gefahrenstellen in der Welt; es gilt die Ladestation zu erreichen ohne in die Gefahrenzone zu geraten. Der Zustand des MDP ist durch die Position des Roboters gegeben. In normalen Positionen kann sich der Roboter in die vier Himmelsrichtungen bewegen; die Gefahrenstellen beschädigen den Roboter irreparabel und es ist keine Bewegung mehr möglich. Diese Bewegungen repräsentiert die möglichen Aktionen des Roboters. Der Roboter bewegt sich jedoch nicht vollständig präzise, eine Bewegung gelingt nur mit einer Wahrscheinlichkeit von 80% fehlerfrei. Mit einer Chance von jeweils 10% weicht der Roboter in eine der orthogonalen Richtungen ab. In Abb. 1 ist eine bildliche Darstellung der Situation zu sehen.

Hiermit ist der mathematische Prozess vollständig definiert, insbesondere die Aktionen und ihre Auswirkungen auf das Gesamtsystem. Jedoch fehlt noch die zweite wichtige Zutat: ein Ziel bzw. die Aufgabe des Roboters. Zum Beispiel könnte man daran interessiert sein, die Ladestation mit maximaler Wahrscheinlichkeit zu erreichen – je nach Position der Gefahrenstellen kann selbst die optimale Wahrscheinlichkeit kleiner als 1 sein. Weiterhin könnte der Energievorrat des Roboters bereits knapp sein und das Ziel ist, die Ladestation in höchstens 10 Schritten zu erreichen. Bereits hier zeigen sich Unterschiede in der optimalen Lösung, eine genauere Analyse dieser Situationen zeigt bereits interessante Dynamiken von MDP auf. Weiterhin ist man oft an dem konkreten „Programm“ interessiert, das die optimale Lösung erreicht, also ein „Rezept“ wie der Roboter in welcher Situation gesteuert werden muss. Diese Rezepte werden oft als *Strategie* bezeichnet.

Mit diesen Grundlagen kann das grundlegende Ziel „(probabilistischer) Verifikation“ definiert werden. Im Wesentlichen beschäftigt sich dieses Feld damit, Systeme wie z.B. MDP zusammen mit einem (formell definierten) Ziel zu analysieren bzw. zu optimieren. Auch

beispielsweise künstliche Intelligenz beschäftigt sich tiefgehend mit diesen Fragestellungen. Im Kontrast zu KI hat Verifikation jedoch als fundamentales Ziel, beweisbar korrekte Ergebnisse zu liefern, während das primäre Ziel von KI üblicherweise ist, ein möglichst gutes Ergebnis ohne Garantien zu finden. Als solches ist Verifikation insbesondere essentiell für kritische Anwendungsgebiete wie Medizin oder Luft- und Raumfahrt. Hier können selbst kleinste Fehler schwerste Folgen mit sich bringen. Die rigorosen Methoden der Verifikation bieten eine verlässliche Basis für die Analyse solcher Prozesse. Insbesondere können potentielle Fehler identifiziert und die Quelle des Problems ausfindig gemacht werden bevor der Fehler im Einsatz des Systems auftritt. Damit ist Verifikation ein wichtiges Forschungsfeld mit mehr und mehr essentiellen Anwendungen.

Wie bereits angedeutet, ist ein wichtiger Teil der Verifikation der exakte Formalismus der verwendet wird, um das Ziel zu beschreiben. Das erste Beispiel (Erreichen der Ladestation) entspricht einem sogenannten *Erreichbarkeits*-Ziel („reachability“). Hier sind wir daran interessiert, gegebene Zustände des Systems mit maximaler (oder minimaler) Wahrscheinlichkeit zu erreichen. Das zweite Beispiel (Erreichen der Ladestation mit einem Zeitlimit) kann bspw. als *Zeit-limitiertes Erreichbarkeits*-Ziel („step-bounded reachability“) oder, etwas genereller, als *Kosten-limitiertes Erreichbarkeits*-Ziel („cost-bounded reachability“) modelliert werden. Jenseits dieser Erreichbarkeitsziele existiert eine enorme Vielzahl an weiteren Zielen. Jedes dieser Ziele bringt eigene Herausforderungen mit sich; die algorithmische Komplexität der assoziierten Entscheidungsprobleme reicht über das gesamte Spektrum von sub-linear bis hin zur Unentscheidbarkeit. Ebenso schwankt die Komplexität der notwendigen Strategien enorm: Während für manche Ziele eine einzelne Aktion pro Zustand ausreicht, benötigen andere sowohl Zufall als auch unendlich viel Speicher, und manchmal existiert gar keine optimale Strategie.

In der Praxis beobachtet man außerdem, dass theoretisch bzgl. der Komplexität optimale Algorithmen oft von „schlechteren“ Varianten um Längen geschlagen werden. Beispielsweise sind manche polynomielle Algorithmen auf vielen konkreten Beispielen aus der Praxis um mehrere Größenordnungen langsamer als manche exponentielle Algorithmen. Damit ist die praktische Analyse der Algorithmen und konkrete, performante Implementierungen ein weiteres wesentliches Ziel in der Verifikation.

Zusammenfassend beschäftigt sich Verifikation also (unter anderem) mit den folgenden vier Problemen: a) Formalismen um Systeme adäquat zu beschreiben (z.B. MDP), b) Formalismen für die zu optimierenden Ziele (z.B. „Erreichen der Ladestation“), c) mathematisch korrekte Algorithmen zur Lösung des resultierenden Entscheidungs- bzw. Optimierungsproblems und d) konkrete Implementierung und Anwendung dieser Algorithmen. Diese Arbeit bringt mehrere Fortschritte zu Punkten b), c) und d), die im Folgenden skizziert werden.

2 Formelle Grundlagen

Bevor die einzelnen Themen in mehr Detail präsentiert werden, wird ein Minimum an formellen Definitionen etabliert. Im Sinne der Lesbarkeit werden präzise Formalismen und detaillierte Verweise weggelassen, diese sind zusammen mit weiteren Kommentaren in der Dissertation zu finden [Me21]. Für einen allgemeinen Einblick in probabilistische Verifikation von MDP wird auf bspw. [BK08; Pu94] verwiesen.

MDP sind ein Tupel $(S, A, \mathsf{Av}, \Delta)$, wobei S eine endliche Menge an *Zuständen*, A eine endliche Menge an *Aktionen*, $\mathsf{Av} : S \to 2^A$ für jeden Zustand die Menge der *verfügbaren Aktionen* beschreibt, und $\Delta : S \times A \to \mathrm{Dist}(S)$ die *Transitionsfunktion* ist. Intuitiv stehen in einem Zustand $s \in S$ die Aktionen $\mathsf{Av}(s)$ zur Verfügung. Wird eine solche Aktion $a \in \mathsf{Av}(s)$ gewählt, bewegt sich das System in einen Nachfolger-Zustand s', der probabilistisch aus der Verteilung $\Delta(s, a)$ gezogen wird. Ausgehend von einem Startzustand $\hat{s}$ kann man diesen Prozess nun beliebig lange wiederholen und erhält somit eine unendliche Sequenz die zwischen Zuständen und Aktionen alterniert. Eine solche Sequenz wird „Pfad“ genannt, ein endliches Präfix eines Pfades wird als *endlicher Pfad* bezeichnet.

Die Entscheidungsfindung in MDP – wann welche Aktion zu wählen ist – wird durch *Strategien* formalisiert. Im Allgemeinen sind Strategien Funktionen die für jeden endlichen Pfad eine Wahrscheinlichkeitsverteilung über die momentan verfügbaren Aktionen liefert. Intuitiv kann man eine Strategie also wie eine Handlungsvorschrift verstehen: Gegeben alle Ereignisse die bis jetzt aufgetreten sind liefert die Strategie den nächsten Schritt. Für eine fixe Strategie sind also alle Entscheidungen fixiert, welchen Pfad das System letztendlich generiert ist nun nur noch eine Zufallsfrage. In anderen Worten liefert ein MDP zusammen mit einer Strategie eine definierte Wahrscheinlichkeitsverteilung über die Menge der Pfade.

Ziele wie z.B. Erreichbarkeit werden formell üblicherweise als Zufallsvariable über die Menge der Pfade definiert. Wie gerade angedeutet ist für eine fixierte Strategie der Pfad des Systems nur noch eine Frage des Zufalls, für jede Strategie erhält man also einen Erwartungswert einer Zufallsvariable usw.

Für ein gegebenes Ziel ist man beispielsweise daran interessiert, den Erwartungswert dieses Ziels über alle Strategien zu optimieren. Hier stellt sich nun eine Vielzahl von Fragen: Gibt es eine optimale Strategie oder kann man den optimalen Wert nur annähern? Wenn ja, welche Struktur hat diese Strategie? Werden zufällig gewählte Aktionen oder „Gedächtnis“ der Strategie benötigt? Welche Komplexität hat das Entscheidungsproblem? Kann man den optimalen Wert effizient approximieren? Mit diesen und weiteren Fragen beschäftigt sich diese Arbeit im Folgenden.

3 Mean Payoff

Mean payoff ist ein „kostenbasiertes“ Ziel. Konkret bedeutet dies, dass mit jeder Aktion des Systems gewisse Kosten assoziiert werden. Formell wird eine Kostenfunktion $r : A \to \mathbb{R}$

gegeben.[3] Diese Kosten können zum Beispiel die Energie die zum Ausführen einer Aktion nötig ist repräsentieren. So kann beispielsweise der durchschnittliche Energiebedarf eines Roboters modelliert werden. Gleichzeitig können Kosten auch als „Erträge“ interpretiert werden. Hier kann z.B. die durch ein Kraftwerk erzeugte Energie modelliert werden.

Um nun zum mean payoff zu gelangen, also die „durchschnittlichen Kosten“, betrachtet man zuerst einen fixen Pfad, also eine unendliche Abfolge von Zuständen und Aktionen. Dieser Pfad induziert eine ebenso unendliche Folge von Kosten (bzw. Erträgen). Die Summe dieser Kosten repräsentiert somit den gesamten Aufwand für diesen Pfad. Im Allgemeinen ist diese Summe unendlich bzw. nicht definiert, also betrachtet man oft stattdessen den Durchschnittswert „im Unendlichen“. Formell ist der mean payoff eines Pfads also definiert als $\liminf_{n\to\infty} \frac{1}{n} \sum_{i=1}^{n} r_i$, wobei r_i die Kosten in Schritt i sind. Nun erhält man also eine Zufallsvariable, die für jeden Pfad die durchschnittlichen Kosten pro Schritt liefert. Die zentrale Frage des mean payoff Problems ist es nun, die erwarteten durchschnittlichen Kosten zu minimieren (bzw. die Erträge zu maximieren).

Dieses Problem wurde schon vor mehreren Dekaden umfangreich behandelt (siehe bspw. [Pu94, Chapter 8 & 9]). Insbesondere existiert ein theoretisch optimaler Algorithmus, der die genannte Frage optimal in polynomieller Zeit mittels eines linearen Programms beantwortet. Jedoch ist dieser Algorithmus in der Praxis nur auf kleinere Systeme mit weniger als 1 Mio. Zuständen realistisch anwendbar. Als populäre Alternativen bieten sich im Kontext von MDP Werte- und Strategieiteration an: Varianten dieser beiden Ansätze existieren für eine Vielzahl an Zielen auf MDP. Oft haben sie zwar exponentielle oder noch schlechtere Komplexität, zugleich sind sie in der Praxis oft wesentlich schneller. Die Dissertation präsentiert effiziente Adaptionen beider Ansätze zum mean payoff Problem.

Für die Werteiteration wird zuerst eine jahrzehntealte, für diesen Ansatz essentielle, Vermutung ([Pu94, Section 9.4.2]) widerlegt und ein alternativer Lösungsweg aufgezeigt. Mit Hilfe dieser Ergebnissen wird ein praktikabler Algorithmus basierend auf Werteiteration definiert. Weiterhin wird dieser grundlegende Ansatz noch einmal mit der Strategie der partiellen Exploration ergänzt: Hier entscheidet der Algorithmus basierend auf mehreren Heuristiken welche Teile des Systems überhaupt konstruiert werden sollen und in welche Regionen des Systems der meiste Berechnungsaufwand investiert werden soll. Obwohl der Algorithmus von Heuristiken gesteuert wird, sind die Resultate dennoch beweisbar korrekt. Basierend auf diesen Ergebnissen verwendet der Strategieiterations-Ansatz Dekomposition und Approximation, um viele seiner Teilschritte zu beschleunigen. Insbesondere ist trotz der Verwendung von Approximationsmethoden das letztendliche Ergebnis exakt.

Diese beiden Methoden sind die ersten, die mean payoff ohne weitere Annahmen effizient berechnen. Insbesondere der Werteiteration-Algorithmus kann dank seiner Heuristiken manche Systeme mit mehreren Milliarden Zuständen in wenigen Sekunden lösen.

[3] Oft werden die Kosten stattdessen für Zustände definiert, dies ist für die Zwecke der Arbeit jedoch äquivalent.

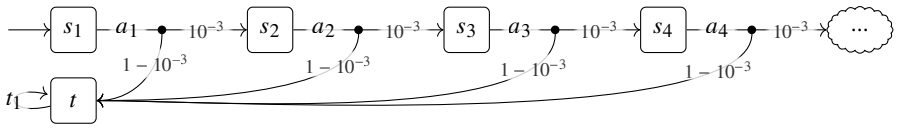


Abb. 2: Beispiel-MDP um Kerne zu motivieren. Zustände werden durch Boxen, Aktionen durch Punkte und probabilistische Transitionen durch Pfeile dargestellt.

4 Lineare Temporallogik

Ähnlich zu mean payoff basiert LTL [Pn77] auf „Beschriftungen“ von Zuständen. Konkret wird hier jedem Zustand des Systems eine Menge an *atomaren Propositionen* aus einer gegebenen Menge AP zugewiesen. Formell geschieht dies durch eine Funktion $l : S \to 2^{\mathsf{AP}}$, 2^{AP} wird oft als „Alphabet“ bezeichnet. Wie zuvor erzeugt jeder Pfad eine unendliche Sequenz an „Buchstaben“ des Alphabets, oft „Wort“ genannt. LTL bietet einen eleganten Formalismus um bestimmte Bedingungen an solche Sequenzen bzw. Worte zu beschreiben. Konkret kann man bspw. mit $\mathsf{AP} = \{\text{empfange}, \text{sende}\}$ modellieren, dass „Immer wenn ein ‚empfange‘ auftritt, muss ein ‚sende‘“ folgen. Ähnlich zum Erreichbarkeit-Ziel kann nun die Menge aller Pfade in jene unterteilt werden, deren entsprechendes Wort die Formel erfüllt, und jene, für die dies nicht gilt (insbesondere ist LTL eine Generalisierung des Erreichbarkeit-Ziels). Für eine gegebene Strategie erhält man also eine Wahrscheinlichkeit die Formel zu erfüllen. Wie zu erwarten ist hier nun das Ziel, diese Wahrscheinlichkeit zu maximieren. Weitere Informationen sind bspw. in [BK08, Section 10.3, 10.6.4] zu finden.

Der klassische *Automatentheoretische Ansatz* [VW86] übersetzt die LTL Formel zu einem sogenannten ω-Automaten, berechnet das *Produkt* zwischen MDP und Automaten, analysiert das Produkt bzgl. *gewinnender Regionen*, und berechnet schließlich die Erreichbarkeit dieser gewinnenden Regionen. Wie zu erwarten beeinflusst die Größe des ω-Automaten die Größe des Produkts und damit die gesamte folgenden Berechnung wesentlich. Insofern ist eine effiziente Übersetzung von LTL zu Automaten ein integraler Teil der Verifikation von LTL-Formeln auf MDP. Unglücklicherweise ist die Übersetzung von LTL zu adäquaten ω-Automaten doppelt exponentiell. Jedoch wurden in den letzten Jahren viele Sonderfälle und Sub-Klassen von LTL identifiziert, für die eine effiziente Übersetzung möglich ist. Also gilt es, eine erweiterbare und effiziente Implementierung zur Verfügung zu stellen.

In diesem Teil der Arbeit wird das Tool `Rabinizer 4.0` sowie die unterliegende Bibliothek und Algorithmensammlung `Owl` präsentiert. Dies fasst eine langjährige Reihe an Forschungsergebnissen rund um LTL zu Automaten Übersetzungen sowie diverse praktische Optimierungen zusammenfassen und öffnet die Türen zu weiterer Forschung. `Owl` wird stetig weiterentwickelt und bietet durch den Fokus auf Offenheit und Wiederverwendbarkeit bereits die Basis für eine Vielzahl an weiteren Forschungsprojekten.

5 Kerne

Als nächstes wird das neu definierte Konzept der *Kerne* eines MDP präsentiert. Diese bieten ein mathematisches Grundgerüst um systematisch „unwichtige“ Teile eines MDP zu identifizieren. Solche unwichtigen Zustände können entfernt bzw. ignoriert werden und somit eine Vielzahl an existierenden Algorithmen direkt beschleunigt werden. Motiviert durch die Ideen von partieller Exploration (wie zum Beispiel für mean payoff) wurde feststellt, dass manche Zustände eines Systems unter keinen Umständen mit nennenswerter Wahrscheinlichkeit erreicht werden können. Für viele Ziele kann man dann beweisen, dass diese Zustände, ohne diese zu analysieren, beweisbar kaum einen Einfluss auf das Gesamtergebnis haben. Beispielsweise könnte, nachdem eine Übertragung 20 mal in Folge fehlgeschlagen ist, die Übertragung beim 21-ten mal erfolgreich sein, jedoch ist die Chance von 20 Fehlern in Folge auf einem adäquaten Medium so gering, dass das genaue Ergebnis dieser Situation im Wesentlichen irrelevant für die gesamte Wahrscheinlichkeit einer erfolgreichen Übertragung ist. Intuitiv kann also nach dem 20-ten Fehler in Folge einfach davon ausgegangen werden, dass die Übertragung sicherlich fehlschlägt. Ein vereinfachtes Beispiel ist in Abb. 2 zu sehen. Hier ist die Wahrscheinlichkeit, die „Wolken“-Region zu erreichen 10^{-12}, also für die meisten Betrachtungen irrelevant. Klassische Methoden würden diese Regionen jedoch genau analysieren – hier schaffen Kerne Abhilfe.

Eine Menge an Zuständen wird als p-Kern bezeichnet, falls die maximale Wahrscheinlichkeit einen Zustand außerhalb des Kerns zu erreichen maximal p ist. In anderen Worten spielt sich die gesamte Entwicklung des Systems mit einer Wahrscheinlichkeit von $1 - p$ innerhalb des Kerns ab. Für kleine p hat also alles außerhalb des Kerns nur marginalen Einfluss, da diese Bereiche in jedem Fall nur mit geringer Wahrscheinlichkeit erreicht werden.

In der Arbeit wird gezeigt, dass das Bestimmen eines kleinen bzw. optimalen Kerns im Allgemeinen NP-vollständig ist. Jedoch wird auch ein effizientes Approximationsverfahren für Kerne gezeigt. Dieser überraschend einfache Algorithmus konstruiert Teile des Systems inkrementell bis ein Kern gefunden wurde anstatt das System zuerst komplett aufzubauen und dann Zustände zu entfernen. Für geeignete Systeme werden hier nur wenige hundert der eigentlichen Milliarden Zuständen konstruiert. Somit kann der Zeit- und Speicherbedarf folgender Analyse um Größenordnungen reduziert. Weiterhin ist der Algorithmus sogar auf manche unendlich große Systeme anwendbar. Außerdem kann der Algorithmus unter anderem durch komplexe Heuristiken aus dem Bereich des Machine Learning unterstützt werden und dennoch mathematisch korrekte Ergebnisse liefern.

Obwohl Kerne anfangs zum Zweck der Optimierung existierender Algorithmen definiert wurden, bieten sie einen neuen, mathematisch greifbaren Formalismus um die „Wichtigkeit“ verschiedener Teile des Systems zu quantifizieren. Dies bereitet den Weg für eine Vielzahl weiterer Anwendungen, insbesondere um Systeme zu *verstehen*, indem wir die wichtigsten Elemente automatisch identifizieren und hervorheben. Beispielsweise sehen wir im vorherigen Beispiel sofort, dass man sich zuerst auf diese wenigen hundert Zustände und deren Zusammenhänge konzentrieren sollte um die wesentliche Gesamtdynamik zu verstehen.

Weiterhin können Kerne als eine Generalisierung der erreichbaren Zustände interpretiert werden: In Systemen ohne stochastische Elemente ist ein Zustand entweder erreichbar oder nicht. Üblicherweise werden nur erreichbaren Zustände in der Analyse berücksichtigt, bzw. oft implizit angenommen, dass nur erreichbare Zustände betrachtet werden. In stochastischen System hingegen ist die „Erreichbarkeit“ ein Kontinuum – ein Zustand kann bspw. mit maximal 50% Wahrscheinlichkeit erreichbar sein. Ein p-Kern entfernt also wenig erreichbare Zustände, insbesondere ist ein 0-Kern genau die Menge aller erreichbarer Zustände. Dieser generelle Blickwinkel überträgt sich natürlich auch auf andere Bereiche und findet bereits jetzt Anwendung in der Analyse von Probabilistischen Programmen.

6 Risikobewusste Analyse

Als letzter Teil folgt ein wesentlicher Beitrag zur *risikobewussten* Analyse von probabilistischen Systemen. Risiko ist ein allgegenwärtiges Konzept und ein elementarer Bestandteil der menschlichen Entscheidungsfindung. Daher haben bereits viele Forschungsbereiche wie Finanzforschung und Psychologie versucht, das menschliche Verständnis von Risiko zu beschreiben und zu quantifizieren. Im Gegensatz dazu wurde Risiko in der Verifikation weitestgehend vernachlässigt, obwohl es gerade in diesem Bereich essentiell ist. Insbesondere werden viele kritische Anwendungen, gerade in der Luft- und Raumfahrt, selten bzw. nur einmalig verwendet. Hier ist das Gesetz der großen Zahlen nicht anwendbar und damit der Erwartungswert als Ziel unter Umständen weitestgehend nutzlos.

Als konkretes Beispiel wird ein abstraktes Modell eines Kraftwerks diskutiert. Auf den ersten Blick scheint es natürlich, die durchschnittliche Leistung des Kraftwerks maximieren zu wollen. (Dies kann beispielsweise mit dem zuvor diskutierten mean payoff Ziel erreicht werden.) Im Beispiel hat das Kraftwerk zwei Betriebsmodi. Im „normalen“ Betriebsmodus produziert das Kraftwerk 100MW, im „Höchstlast“-Modus 110MW. In letzterem Modus besteht außerdem eine 5% Chance, das Kraftwerk temporär zu beschädigen und 0MW Leistung zu erzielen. Die „optimale“ Strategie für maximale durchschnittliche Leistung ist also, das Kraftwerk permanent auf Höchstlast laufen zu lassen und das Risiko eines Totalausfalls willentlich einzugehen. Dies ist in diesem Szenario womöglich nicht wünschenswert – jedoch hängt das natürlich auch vom Kontext und der Modellierung ab.

Als erste Reaktion könnte man versuchen, dieses Problem mithilfe von „Strafen“ zu beheben. Beispielsweise könnte ein Totalausfall als „-100MW“ modelliert werden. Dies zieht jedoch eine Vielzahl an Problemen mit sich. Zum einen ist das Endergebnis kaum interpretierbar: Falls der optimale Wert bspw. „50MW“ sein sollte, ist nicht ersichtlich ob das Kraftwerk durchgehend 50MW produziert oder zu 75% der Zeit 100MW und zu 25 % ausfällt, was einem tatsächlichen Schnitt von 75MW entspräche. Weiterhin ist nicht ersichtlich, wie hoch die Strafen modelliert werden sollten. Sind sie zu niedrig, werden sie einfach in Kauf genommen, sind sie zu hoch, ist die optimale Strategie potentiell zu risikoavers. Konkret könnte dies bedeuten, dass die optimale Strategie das Kraftwerk niemals in Betrieb nimmt.

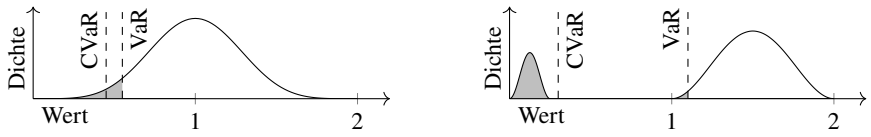


Abb. 3: Illustration von VaR und CVaR für zwei Zufallsvariablen.

Dies zeigt ein wichtiges Problem der Verifikation und anderer formeller Ansätze auf: Da das Ziel ist, die mathematisch bestmögliche Leistung zu erreichen, werden alle anderen, nicht konkret spezifizierten Ziele komplett außer Acht gelassen. Insbesondere wird ein beliebig hohes Risiko eingegangen um eine marginale Verbesserung zu erreichen. Dies ist jedoch kein fundamentales Problem – es bedarf lediglich einer Möglichkeit, alle gewünschten Ziele zu spezifizieren. Als solches gibt es ein großes Interesse an „risikobewussten“ Ansätzen, die das Risiko auf einem akzeptablen Niveau halten, anstatt es entweder komplett zu ignorieren oder um jegliche Kosten versuchen es vollständig zu eliminieren.

Als ersten Schritt in diese Richtung präsentieren wir *conditional value-at-risk* (CVaR) für MDP mit mean payoff zusammen mit einer ausführlichen Behandlung der Theorie. CVaR ist ein klassisches, in Finanztheorie und OR etabliertes Risikomaß mit vielen wünschenswerten Eigenschaften [RU00]. Risikomaße sind, genauso wie der Erwartungswert, eine „Aggregation“ einer Zufallsvariable. Um jedoch das Konzept von CVaR genauer zu erklären, muss zuerst das verwandte Risikomaß *value-at-risk* (VaR) definiert werden. Sowohl CVaR als auch VaR haben einen Parameter $p \in [0, 1]$. Die VaR einer Zufallsvariable X ist dann ein Wert v so dass X mit einer Wahrscheinlichkeit von p einen Wert von maximal v hat, VaR ist also das schlechteste p-Quantil von X. Eine grafische Darstellung ist in Abb. 3 zu sehen. Dieser Wert soll die Frage „Was ist ein üblicher Problemfall?“ beantworten. Wie in der Abbildung angedeutet ignoriert VaR jedoch Ausreißer und verhält sich Änderungen in X gegenüber nicht stetig. Um diese Probleme zu beheben, betrachtet CVaR den Erwartungswert aller „schlechten“ Fälle, also aller die schlechter als VaR sind. Somit beantwortet CVaR die Frage „Was kann man von einem Problemfall erwarten?“ und berücksichtigt alle potentiellen Probleme ohne übermäßig pessimistisch zu sein. Insbesondere erlaubt CVaR zwischen Worst-Case Analyse ($p = 0$) und regulärem Erwartungswert ($p = 1$) zu interpolieren.

Um CVaR auf MDP zu übertragen ersetzt man in den präsentierten Zielen den Erwartungswert der entsprechenden Zufallsvariable durch CVaR. Beispielsweise kann man somit die CVaR eines mean payoff Ziels optimieren. In der Arbeit wird der theoretische Aspekt dieser Optimierung genau diskutiert. Insbesondere wird auf die Struktur der optimalen Strategien sowie die algorithmische Komplexität eingegangen. Der präsentierte Ansatz ist außerdem in der Lage, die Erwartung eines mean payoff Ziels zu maximieren und zeitgleich CVaR und VaR für weitere mean payoff Ziele zu kontrollieren. Als solches kann man also die Leistung eines Systems unter gewissen Risikogrenzen optimieren. Noch dazu ist die algorithmische Komplexität in diesem Fall genau so hoch wie reine Erwartungsoptimierung. Es gibt also keinen fundamentalen Grund, Risiko außer acht zu lassen.

7 Schluss

Die Arbeit zeigt vielfältige und signifikante Ergebnisse in Grundlagenbereichen der Informatik und sind sowohl theoretischer als auch praktischer Natur. Während die Ergebnisse zu mean payoff konkrete Anwendungsfälle beschleunigen, bereitet `Owl` anderen Forschern die Möglichkeit, Ideen rund um LTL schnell zu implementieren und zu evaluieren. Das grundlegend neue Konzept der Kerne kann neben effizienteren Algorithmen insbesondere in Anwendungsgebieten wie der Systembiologie dazu beitragen, komplexe Systeme greifbar zu machen. Die Betrachtung von Risiko ist ein essentieller Teil der Analyse sicherheitskritischer Systeme, für die durch diese Arbeit ein wichtiger Grundstein gelegt wurde.

Literatur

[BK08] Baier, C.; Katoen, J.: Principles of model checking. MIT Press, 2008.

[Me21] Meggendorfer, T.: Verification of Discrete-Time Markov Decision Processes, Diss., Technical University of Munich, Germany, 2021.

[Pn77] Pnueli, A.: The Temporal Logic of Programs. In: 18th Annual Symposium on Foundations of Computer Science. S. 46–57, 1977.

[Pu94] Puterman, M. L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley, 1994, ISBN: 978-0-47161977-2.

[RU00] Rockafellar, R. T.; Uryasev, S.: Optimization of conditional value-at-risk. Journal of risk 2/, S. 21–42, 2000.

[VW86] Vardi, M. Y.; Wolper, P.: An Automata-Theoretic Approach to Automatic Program Verification (Preliminary Report). In. IEEE Computer Society, S. 332–344, 1986.

[Wh93] White, D. J.: A survey of applications of Markov decision processes. Journal of the operational research society 44/11, S. 1073–1096, 1993.

Tobias Meggendorfer wurde am 20. August 1993 in Landshut geboren. Seit 2002 besuchte er das Hans-Leinberger-Gymnasium in Landshut und begann im Wintersemester 2008 parallel das Vorstudium für herausragende Schüler „Schüler.IN.TUM“. Nach einem ausgezeichneten Abitur im Jahr 2011 (Schnitt 1.3) begann er nahtlos mit einem Mathematik-Studium im Rahmen des „twoinone“ Programms an der TUM. Im Winter 2015 schloss er den Master in Mathematik erfolgreich ab (Schnitt 1.2) und schloss direkt ein Masterstudium Informatik an. Im folgenden Jahr begann er parallel zum Master seine Promotion und stellte beide Abschlussarbeiten im Sommer 2020 fertig (Informatik Schnitt 1.1). Durch die Corona-Pandemie verzögerte sich die Verteidigung der Promotion bis zum Februar 2021. Direkt im Anschluss trat Herr Meggendorfer seine Post-Doc Stelle an der IST Austria im März 2021 an, wo er weiterhin tätig ist.

Tiefe Netzwerke, die wissen, wenn sie etwas nicht wissen[1]

Alejandro Molina Ramirez[2]

Abstract: Viele Modelle des Maschinellen Lernens konzentrieren sich darauf, die größtmögliche Genauigkeit zu bieten. Allerdings sind diese Modelle meist Blackboxen, die schwer zu untersuchen sind. Es handelt sich oft um diskriminative Modelle, die Ergebnisse anhand von Trainingsdaten erzielen, aber kein eigenes Modell für die Daten entwickeln. Wir wollen jedoch wissen, ob sich die aktuellen Daten, die das Modell verarbeitet, mit den Trainingsdaten decken. Mit anderen Worten: Ist es qualifiziert genug, Entscheidungen zu treffen und "weiß es, wovon es redet"? Diese Dissertation [Mo21] konzentriert sich auf tiefe generative Modelle, die auf probabilistischen Schaltkreisen basieren; die es uns erlauben, eine breite Spanne von normalisierten Wahrscheinlichkeitsanfragen in einer garantierten Rechenzeit zu beantworten. Diese generativen Modelle können dann auf Bias überprüft werden, einschließlich, wie sicher sie sich einer bestimmten Antwort sind, weil sie "wissen, wenn sie es nicht wissen".

1 Einleitung

Maschinelles Lernen wird zunehmend für das Treffen weitreichender Entscheidungen eingesetzt, d. h. Entscheidungen, die erhebliche Auswirkungen auf den Einzelnen haben können. Wir erwarten, dass Modelle des Maschinellen Lernens sowohl genau als auch fair sind. Es ist daher auch zwingend erforderlich, zu prüfen, ob die genutzten Modelle für die jeweilige Aufgabe geeignet sind. Leider signalisieren die Modelle selbst nicht, wenn sie für bestimmte Aufgaben ungeeignet sind.

Wir sind nicht nur an Endergebnissen interessiert, sondern auch am Verstehen der Daten, die zu den Ergebnissen führen. Wir beginnen mit der Spezifikation eines Rahmenmodells für die Erstellung, Optimierung und Interaktion von Modellen, die dazu in der Lage sind. Der sinnvollste Ansatz ist hier das Verwenden von statistischen Modellen und probabilistischen Schlussfolgerungen, da sie einen klaren Rahmen bieten, der zu diesen Bedürfnissen passt. Für eine ausführliche Diskussion siehe [Pe14].

Probabilistische Verfahren unterscheiden zwischen verschiedenen Arten von Modellen. Die prädiktiven Modelle sind oft als diskriminative Modelle bekannt. Im Gegensatz dazu werden Modelle, die sich nicht nur auf die Vorhersage, sondern auch auf das Erstellen einer Repräsentation der Daten fokussieren, generative Modelle genannt.

Bei den diskriminativen Modellen erhalten wir eine bedingte Wahrscheinlichkeitsverteilung $P(\mathbf{Y}|\mathbf{X})$ für die Labels $\mathbf{Y}$, gegeben die Merkmale $\mathbf{X}$. Bei diesen Modellen erhalten wir

[1] Englischer Titel der Dissertation: "Deep Networks That Know When They Don't Know"

[2] Amazon.com, Seattle, USA, ale@molina.ai

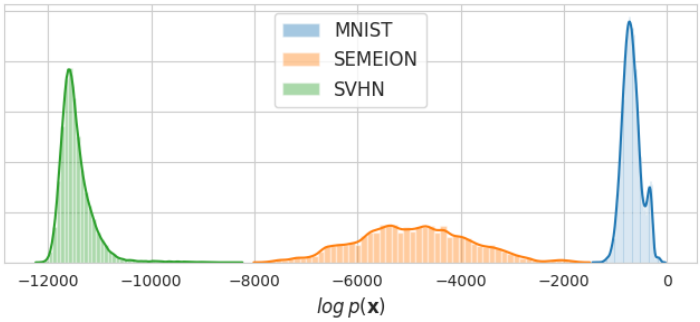


Abb. 1: Histogramme der Log-Wahrscheinlichkeiten von Bildern in den Testdatensätzen MNIST, SEMEION und SVHN. Das Modell wurde auf MNIST-Trainingsdaten trainiert. MNIST ist hier der Datensatz, den das Modell "kennt", während die beiden anderen Datensätze zur Kategorie der "nicht bekannten" Daten gehören. Beachtenswert ist, dass die Histogramme sich nicht überschneiden.

nur die Vorhersagen für $\mathbf{Y}$, da sie keine zusätzlichen Informationen über die Eingaben $\mathbf{X}$ speichern. Wenn wir das Modell nach einer Eingabe fragen, die zu einer unbekannten Klasse gehört, sollten wir eine Antwort mit maximaler Entropie erhalten, d. h. einen einheitlichen Wahrscheinlichkeitswert für beide Klassen. Leider ist dies in der Praxis selten der Fall.

Die generativen Modelle hingegen erstellen eine multivariate Verteilung für die Labels $\mathbf{Y}$ und die Merkmale $\mathbf{X}$, um $P(\mathbf{X}, \mathbf{Y})$ zu erhalten. Hier erfahren wir sowohl etwas über die Vorhersagen $\mathbf{Y}$ als auch über die Eingabevariablen $\mathbf{X}$. Hätten wir das ideale Modell $P^*(\mathbf{X}, \mathrm{Y})$ zur Verfügung, könnten wir die Unsicherheit in Bezug auf die Eingabe $\mathbf{X}$ über die Randverteilung schätzen $P^*(\mathbf{X}) = \sum_y P^*(\mathbf{X}, \mathrm{Y} = y)$. Hier weist eine geringere Wahrscheinlichkeit im Vergleich zum Trainingsdatensatz darauf hin, dass das Modell die Eingabe nicht "kennt"; z. B. ein Bild von einem Baum in einem Modell, was darauf trainiert ist, zwischen Bildern von Hunden oder Katzen zu unterscheiden. Allerdings haben wir keinen Zugang zu $P^*(.)$. Daher verwenden wir ein generatives Modell mit Marginalisierungsfähigkeiten $P_\theta(\mathbf{X}, \mathrm{Y})$, um uns anzunähern, d. h. $P_\theta(\mathbf{X}, \mathrm{Y}) \simeq P^*(\mathbf{X}, \mathrm{Y})$. In dieser Arbeit, entwickeln wir solche Modelle. Wie wir in Abb. 1 sehen können, kann das Modell eindeutig Daten von "bekannten" und "unbekannten" Quellen unterscheiden, d. h. signalisieren, wenn es etwas nicht "weiß". Darüber hinaus erweitert die Marginalisierungsfähigkeit die verfügbaren Anwendungsmöglichkeiten erheblich. Sie erlaubt es uns, jedes beliebige Attribut zu schätzen, nicht nur die Vorhersagen. Wir sind in der Lage, das Modell auf Verzerrungen zu prüfen, indem wir Paare von marginalen Zufallsvariablen vergleichen, z. B. Kreditbewilligungen in Abhängigkeit vom Alter. Hierzu können einfache Was-wäre-wenn-Fragen verwendet werden, um das Modell zu untersuchen und Vertrauen in es zu entwickeln, oder auch, um die Daten besser zu verstehen.

2 Hintergrund

Wie bereits erwähnt, sind wir an der Schätzung von Wahrscheinlichkeitsverteilungen interessiert, die die gesamte Komplexität der Daten erfassen und dennoch rechnertechnisch lösbar (tractable) sind. Dies ist bekanntlich schon eine Herausforderung. Hier jedoch verlangen wir zusätzlich, dass das Modell auch exakt und tractable für die Marginalisierung sein muss. In dieser Arbeit konzentrieren wir uns auf Sum Product Networks [PD11], eine Instanziierung probabilistischer Schaltkreise. Dies sind tiefe probabilistische graphische Modelle (DPGMs), die in der Lage sind, multivariate Verteilungen darzustellen. Im Gegensatz zu herkömmlichen graphischen Modellen können SPNs *exakte* Inferenz für eine Reihe von Abfragen in einer Zeit durchführen, die *linear* zur Größe des Netzwerks ist, einschließlich Marginalisierung. Diese Aufgaben sind für klassische probabilistische Modelle meist NP-schwer. SPNs hingegen bleiben tractable, mit dem Nachteil, dass sie exponentiell größer sein können.

Definition: Sum Product Networks sind probabilistische graphische Modelle, die eine gemeinsame multivariate Wahrscheinlichkeitsverteilung über die Menge der Zufallsvariablen $\mathbf{X}$ kodieren, d.h. $P(\mathbf{X}) = S(X)$. $G = (\mathbb{N}, \mathbb{E})$ sei ein gerichteter, azyklischer Berechnungsgraph mit *Summen-*, *Produkt-* und *Blattknoten* $\mathbb{N}$. Die Kanten $\mathbb{E}$ geben die Reihenfolge an, in der die Berechnungen durchgeführt werden. Nur die Kanten der *Summenknoten* sind gewichtet.

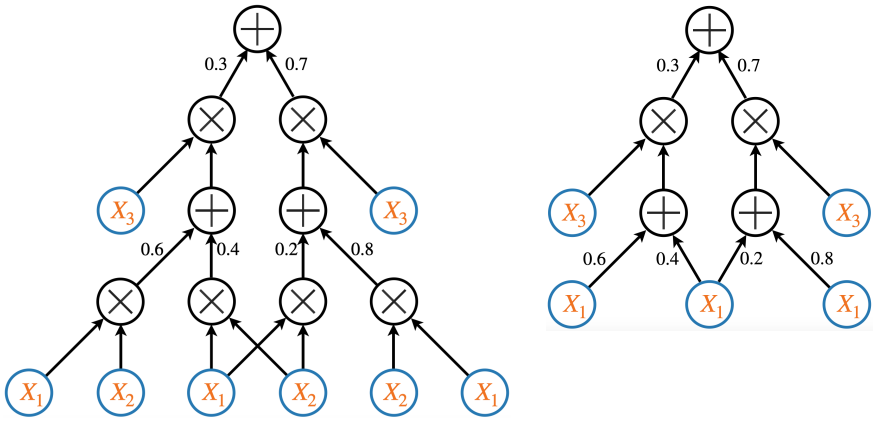
Die drei verschiedenen Knotentypen haben eine probabilistische Semantik:

- Die *Blattknoten* sind beliebige univariate Verteilungen $P(\mathrm{X})$.
- *Summenknoten* kodieren eine Mischung von Verteilungen $P(\mathbf{X}) = \sum_i w_i P_i(\mathbf{X})$.
- *Produktknoten* faktorisieren Unabhängigkeiten $P(\mathbf{X}) = \prod_{i \in I} P_i(\mathcal{X}_i)$.

Inferenz und Marginalisierung: Nachdem wir nun eine Definition haben, wollen wir uns ein Beispiel für eine Inferenz unter Verwendung des SPN aus Abb. 2 ansehen. Um die Wahrscheinlichkeiten in einem SPN zu berechnen, berechnen wir die Werte der Knoten, beginnend bei den Blättern. Wir geben die Nutzerdaten in die Blätter, welche univariate Verteilungen sind, und berechnen so die Wahrscheinlichkeiten der Blätter. Dann werten wir den Graphen von unten nach oben aus: Bei Produktknoten multiplizieren wir die Werte der Kinderknoten, bei Summenknoten summieren wir die gewichteten Werte der Kinderknoten. Der Wert an der Wurzel gibt die Wahrscheinlichkeit für die gegebene Anfrage an. Zur Berechnung der Marginale (Randwerte), d. h. der Wahrscheinlichkeit von Teilmengen der Zufallsvariablen im SPN, setzen wir die Wahrscheinlichkeit an den Blättern für diese Variablen auf 1 und verfahren dann wie zuvor.

Bei einer Wahrscheinlichkeitsabfrage mit vollständiger Evidenz, d. h. dem Fall bei dem wir Zuordnungen zu allen Zufallsvariablen haben, berechnet der Berechnungsgraph die Wahrscheinlichkeit: $P(\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3) = 0.3(P_1(\mathrm{X}_3)(0.6P_3(\mathrm{X}_1)P_4(\mathrm{X}_2) + 0.4P_5(\mathrm{X}_1)P_6(\mathrm{X}_2)) + 0.7(P_2(\mathrm{X}_3)(0.2P_5(\mathrm{X}_1)P_6(\mathrm{X}_2) + 0.8P_7(\mathrm{X}_1)P_8(\mathrm{X}_2))$.

Hier sehen wir, dass die Verbundwahrscheinlichkeit genau berechnet wird, ohne dass wir auf zufällige Stichproben zurückgreifen müssen. Außerdem sind alle Wahrscheinlich-



(a) Struktur eines Sum Product Network (SPN)

(b) SPN nach Marginalisierung X_2

Abb. 2: (links) Ein Sum Product Network kodiert eine multivariate Verteilung über die Zufallsvariablen $P(\mathrm{X}_1, \mathrm{X}_2, \mathrm{X}_3)$. *Summenknoten* sind Mischungen mit festen normalisierten Gewichten. *Produktknoten* zerlegen die Verteilungen nach Unabhängigkeiten. *Blattknoten* sind univariate Verteilungen. (rechts) Das Sum Product Network von links nach Marginalisierung der Zufallsvariablen X_2. Beachtenswert ist, dass die Marginalisierung eine Bearbeitung des Graphen ist, bei der wir die Knoten der Zufallsvariablen entfernen, die wir marginalisieren wollen. Die Modellbewertung folgt der Richtung der Kanten.

keitsabfragen in SPNs normalisiert. Des Weiteren haben wir den Graphen nur einmal durchlaufen, um all diese Berechnungen zu erhalten. Wir können auch das Marginalisierungsverfahren durchführen, indem wir alle Knoten aus dem Graphen entfernen, die mit den zu marginalisierenden Zufallsvariablen assoziiert sind. Abb. 2b zeigt ein Beispiel der Marginalisierung.

Jede bedingte Wahrscheinlichkeitsabfrage $P(\mathbf{Q}|\mathbf{E})$ kann berechnet werden. Dazu berechnen wir erst $P(\mathbf{Q}, \mathbf{E})$ und die Marginalisierung $\sum_q P(\mathbf{Q} = q, \mathbf{E})$ und schließlich erhalten wir mit der Kettenregel die bedingte Wahrscheinlichkeit $P(\mathbf{Q}|\mathbf{E}) = P(\mathbf{Q}, \mathbf{E}) / \sum_q P(\mathbf{Q} = q, \mathbf{E})$.

Um SPNs zu erstellen, können wir mit einem beliebigen SPN-Graphen beginnen und dann eine Verlustfunktion optimieren, um die Gewichte und Parameter zu erhalten. Alternativ dazu können wir gierige Algorithmen verwenden, die die Daten in Zeilen (*Summenknoten*) und Spalten (*Produktknoten*) aufteilen und dann univariate Verteilungen anpassen, wenn keine Aufteilungsoperationen mehr anstehen. Für weitere Details zur Optimierung, zur Berechnung wahrscheinlichster Werte, zur Erstellung von zufälligen Stichproben und zu den Beweisen siehe die Dissertation [Mo21].

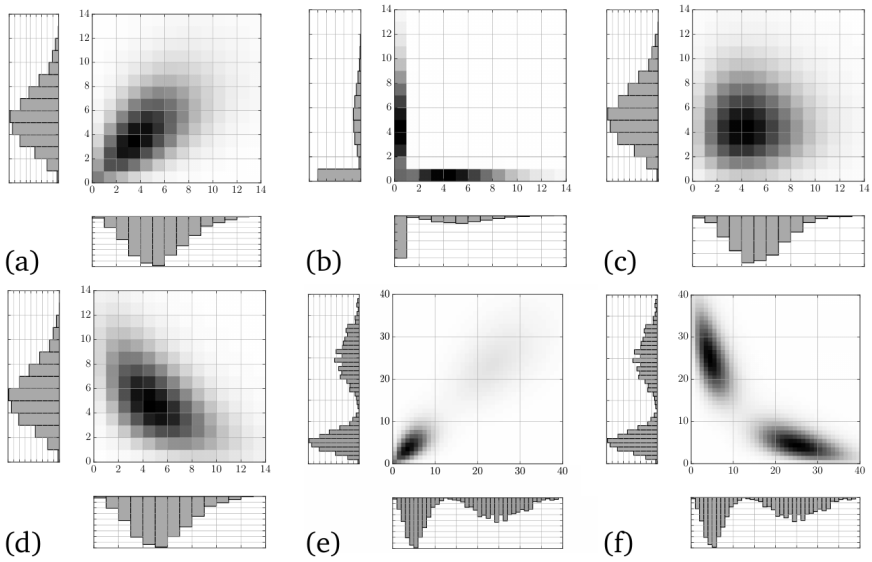


Abb. 3: PSPNs sind flexible multivariate Poisson-Verteilungen. Dargestellt sind die Wahrscheinlichkeiten und Randverteilungen von sechs 2D-Poisson-SPNs mit unterschiedlichen Abhängigkeiten zwischen den beiden Zählvariablen. (a) Positive Abhängigkeit bedeutet, dass zwei Ereignisse häufig zusammen auftreten, (b) negative Abhängigkeit, dass zwei Ereignisse selten zusammen auftreten, (c) Null-Abhängigkeit, dass die Ereignisse nicht korreliert sind, und (d) Anti-Abhängigkeit, dass wenn ein Ereignis eintritt, das andere nicht eintritt. Gemisch mit positiver Abhängigkeit (e) und Anti-Abhängigkeit (f).

3 Beitrag der Dissertation

Poisson Sum Product Networks (PSPNs). Multivariate Zähldaten sind in der Wissenschaft und in vielen anderen Geschäfts- und Alltagssituationen allgegenwärtig. Basierend auf SPNs stellen wir ein neues graphisches Poisson Modell vor, genannt Poisson Sum Product Networks (PSPNs) [MNK17]. Sie repräsentieren multivariate Poisson-Verteilungen, mit allen oben beschriebenen Eigenschaften der Inferenz und Marginalisierung. PSPNs können als eine tiefe Kombination von multivariaten Poisson-Mischmodellen und Merkmalshierarchien angesehen werden. Darüber hinaus ermöglichen sie ein breites Spektrum an Abhängigkeitsinteraktionen zwischen den Zufallsvariablen, wie in Abb. 3 zu sehen ist. Wir präsentieren Algorithmen zum Lernen von PSPNs und zur Berechnung von informationstheoretischen Maßen wie Entropie, Transinformation/Synentropie und Maßen für die Unterschiedlichkeit von Zählvariablen, ohne auf Approximationen zurückgreifen zu müssen. Außerdem zeigen wir eine Verbindung zwischen PSPN und dem Themenmodell Latent Dirichlet Allocation (LDA), das die Struktur von Baum PSPNs mit einer Hierarchie von Themen verbindet. Die experimentellen Ergebnisse mit mehreren synthetischen und realen Datensätzen zeigen, dass PSPN den Stand der Technik übertreffen und dabei tractable bleiben.

Mixed Sum Product Networks (MSPNs). Eine logische Folgefrage ist, ob wir unter Beibehaltung der Tractability-Garantien eine universelle Verteilung für jegliche Art von Daten erstellen können. Hier sprechen wir dann über Modelle für hybride Domänen. Um zu verstehen, warum eine solche universelle Verteilung gebraucht wird, reicht ein Blick auf die Erfolge, die das Maschinelle Lernen in den letzten Jahren erzielt hat und die ständig wachsende Zahl von Disziplinen, die sich darauf stützen. Daten sind heute allgegenwärtig und es ist essentiell, Daten zu verstehen, probabilistische Modelle zu erstellen und mit ihnen Vorhersagen zu treffen.

In den meisten Fällen hängt dieser Erfolg jedoch entscheidend von den Fähigkeiten der Datenwissenschaftler*innen ab, die für die Daten richtige parametrische Form des probabilistischen Modells und den passenden Algorithmus auszuwählen und schließlich die Inferenz durchzuführen. Dies geht oft über die Fähigkeiten von Nicht-Expert*innen hinaus, insbesondere in hybriden Domänen, die aus gemischten – kontinuierlichen, diskreten und/oder kategorischen – statistischen Typen bestehen. Der Aufbau eines probabilistischen Modells, das sowohl aussagekräftig genug ist, um komplexe Abhängigkeiten zwischen Zufallsvariablen verschiedener Typen zu erfassen und gleichzeitig ein effektives Lernen und effiziente Inferenz ermöglicht, ist ein anspruchsvolles Problem. Um die Schwierigkeiten der gemischten probabilistischen grafischen Modellierung zu überwinden, führen wir Mixed Sum Product Networks (MSPNs) ein [Mo18]. Sie sind generalistische gemischte probabilistische Modelle, die komplexe Interaktionen zwischen Zufallsvariablen in hybriden Domänen erfassen können. Hier verwenden wir stückweise polynomiale Verteilungen als Blätter zusammen mit einer neuartigen nicht-linearen, nicht-parametrischen Partitionierung unter Verwendung des Hirschfeld-Gebelein-Rényi Maximalen Korrelationskoeffizienten. MSPNs sind das erste automatische Werkzeug, um multivariate Verteilungen über hybride Domänen zu lernen, ohne dass die Benutzer*innen a-priori die parametrische Form von Zufallsvariablen oder deren Abhängigkeiten bestimmen müssen. Des Weiteren ermöglichen sie es, komplexe probabilistische Abfragen effizient zu beantworten, die zuvor mit klassischen graphischen Modellen nicht durchführbar waren.

Um die Schwierigkeiten der gemischten probabilistischen grafischen Modellierung zu überwinden, führen wir Mixed Sum Product Networks (MSPNs) ein [Mo18]. Sie sind generalistische gemischte probabilistische Modelle, die komplexe Interaktionen zwischen Zufallsvariablen in hybriden Domänen erfassen können. Hier verwenden wir stückweise polynomiale Verteilungen als Blätter zusammen mit einer neuartigen nicht-linearen, nicht-parametrischen Partitionierung unter Verwendung des Hirschfeld-Gebelein-Rényi Maximalen Korrelationskoeffizienten.

Automatic Bayesian Density Analysis (ABDA). Eine weitergehende Frage ist, ob wir parametrische Wahrscheinlichkeitsverteilungen verwenden können, ohne die Flexibilität nicht-parametrischer Modelle zu verlieren. Oder anders ausgedrückt, können wir automatisch Modelle mit Verteilungsannahmen erstellen können, ohne dass Expert*innen hinzugezogen werden müssen? Die nicht-parametrische Natur von MSPNs bedeutet, dass sie möglicherweise zu flexibel sind und die Daten überanpassen könnten. Gäbe es jedoch Vertei-

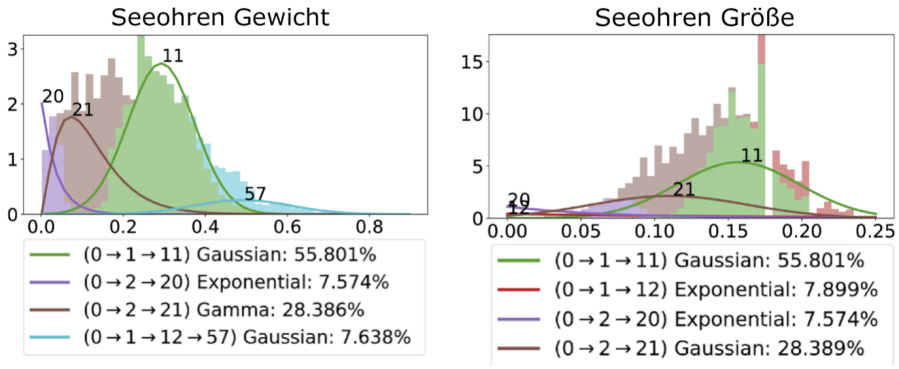


Abb. 4: Mustererkennung mit ABDA. Die Dichten gehören zu einer Partition $\mathbf{X}^{\mathsf{N}}$. Hierarchien beginnen an der Wurzel (z. B. bedeutet $0 \rightarrow 1 \rightarrow 11$, dass die Partition #11 für die grünen Dichten verantwortlich ist). Dann ergibt sich die Regel $\mathcal{P}_1 : 0.15 \leq$ Gewicht $< 0.42 \wedge 0.08 \leq$ Größe < 0.22, mit der Relevanz $(supp(\mathcal{P}_1) = 0.5)$, die uns sagt, dass Größe und Gewicht korrelieren. Beachtenswert ist, dass in der rechten Abbildung fehlerhafte Daten eine Lücke im Histogramm hinterlassen. ABDA identifiziert dennoch korrekt die richtige Verteilung.

lungsannahmen wäre das Modell gezwungen, einen Kompromiss zwischen den Verteilungen und den Daten zu finden. Der Vorteil davon wäre, dass sich das Rauschen in den Daten weniger auf das Gesamtmodell auswirkt. Zusätzlich kann ein solches Modell auch direkt Anomalien oder interessante Muster in den Daten finden. In diesem Zusammenhang haben wir die Automatic Bayesian Density Analysis (ABDA) eingeführt [Ve19]. Sie automatisiert für die gegebenen Daten die Auswahl des geeigneten parametrischen Wahrscheinlichkeits-Modells (bspw. Gauß, Gamma, Poisson oder multinomial) und modelliert die gesamte multivariate Verteilung inklusive der Wechselwirkungen zwischen den Zufallsvariablen. Um den Prozess robuster zu machen, verwenden wir Bayessche Inferenz.

Da ABDA robust gegenüber Ausreißern, fehlenden und fehlerhaften Werten ist, können wir es zum Erkennen von Anomalien nutzen. Während des Trainings werden anomale Elemente tendenziell niedrig gewichteten Mischungskomponenten (*Mikro-Cluster*) oder den Verteilungsenden eines Likelihood-Modells zugeordnet.

Darüber hinaus kann ABDA den Benutzer*innen *lokale* Muster zur Verfügung stellen, in Form von Abhängigkeiten innerhalb einer Datenpartition $\mathbf{X}^{\mathsf{N}} \subseteq \mathbf{X}$, die mit einem beliebigen Knoten verbunden ist. Für jedes Blatt und jedes Wahrscheinlichkeits-Modell kann man ein Muster der Form $\mathcal{P} : \pi_l^d \leq X^d < \pi_h^d$ extrahieren. Diese Technik ist verwandt mit der relativen Häufigkeit in der Assoziationsanalyse, dessen binäre Muster hier verallgemeinert werden, um sie für kontinuierliche und (nicht-binäre) diskrete Zufallsvariablen zu nutzen. Ein solches Beispiel sehen wir in Abb. 4. Wir können zeigen, dass ABDA den aktuellen Stand der Technik für verschiedenen Aufgaben und Szenarien übertrifft, in denen Fachleute die explorative Datenanalyse von Hand durchführen würden. Das Modell kann schnell jede

Frage zu den Daten beantworten, z. B. die Wahrscheinlichkeit verschiedener Szenarien, alle Formen von Vorhersagen, Anomalien in den Daten und mehr. Diese vollständige Automatisierung der Modellierung und Datenanalyse leistet einen großen Beitrag zur Idee eines "Automatisierten Statistikers". Für die nächsten Beiträge lassen wir uns weiter von tiefen neuronalen Netzen inspirieren.

Conditional Sum Product Networks (CSPNs). Tiefe neuronale Netze (DNNs) sind einer der größten Erfolge im Bereich Künstliche Intelligenz. Sie bestehen aus mehreren Schichten, die nacheinander so verbunden sind, dass sie immer komplexere Merkmale der Daten erkennen und repräsentieren können. Diese Netzwerke lernen anhand von Backpropagation (Fehlerrückführung), was tiefes Lernen flexibler macht als flaches Lernen, welches keine höheren Merkmale ableitet. Durch diese Fähigkeit sind sie sehr gut in der Lage, bedingte Verteilungen zu modellieren. Wenn wir mehrere vor-trainierte Netze nutzen oder multimodale Probleme lösen wollen, ist es möglich, mehrere DNNs zu kombinieren, um noch aussagekräftigere Modelle zu erhalten. Multimodale Probleme erfordern die Nutzung verschiedener Datentypen, zum Beispiel Texte, Bilder und Audiodaten und wir möchten für jeden dieser Typen das fortgeschrittenste und spezialisierteste Modell nutzen. Um diese Netze auf eine probabilistische Weise zu kombinieren, werden bisher eher einfache Ansätze genutzt, wie die Mean-Field-Näherung – die nichts anderes ist als Unabhängigkeiten – und/oder die Mischung von Experten.

Für ein besseres Verfahren mehrere Modelle zu kombinieren, erleichtern wir die strukturellen Restriktionen von Sum Product Networks und ändern sie, um multivariate bedingte Verteilungen modellieren zu können. Das Ergebnis nennen wir Conditional Sum Product Networks (CSPNs) [Sh20]. CSPNs ermöglichen eine hierarchische Faktorisierung, welche die Interaktion von bedingten Zufallsvariablen genauer erfasst. CSPNs können Modelle erzeugen, die die Daten gut repräsentieren, selbst wenn sie einfache lineare Modelle kombinieren. Wir zeigen, wie CSPNs verwendet werden können, um autoregressive Modelle zu erstellen und wie die Kombination von tiefen neuronalen Netzen oft den Stand der Technik bei verschiedenen Aufgaben übertrifft. All dies erreichen wir unter Beibehaltung der vorteilhaften Eigenschaften der Traktabilität und Marginalisierung, die wir zuvor beschrieben haben.

Random and Einsum Networks. Schließlich stellen wir zwei Ansätze [Pe19], und [Pe20] vor, die unsere SPNs aus der Perspektive von Berechnungsgraphen betrachten. Inspiriert von der Idee sehr großer Deep-Learning-Modelle wollen wir die Grenzen von SPNs erweitern, indem wir die Größe der Netzwerke und damit die Anzahl der Parameter massiv erhöhen. Wir erstellen Zufallsregionsgraphen, die alle strukturellen Anforderungen von SPNs erfüllen. Wir nutzen dann die Vorteile von hochoptimierten Deep Learning Frameworks, fügen dem Summenknoten Drop-out hinzu, und erweitern die Verlustfunktion, um sowohl diskriminative als auch generative Komponenten einzubeziehen. Schließlich trainieren wir unser Modell mit automatischer Differenzierung und Backpropagation, wie im Fall der Erweiterung mit einem tiefen neuronalen Netzwerk.

Da wir erfreulicherweise auf jeder Stufe des Graphen gültige Verteilungen haben, können wir Optimierer verwenden, die auf unsere Modelle zugeschnitten sind. Insbesondere der Ansatz des Erwartungs-Maximierungs-Algorithmus und seine Batch-Variante sind leicht auf Basis der automatischen Differenzierung zu implementieren. Schließlich führen wir Sum Product Networks ein, bei denen die Inferenz und die Optimierung durch Nutzen der einsteinschen Summenkonvention weiter beschleunigt wird. Diese Notation ermöglicht es uns, Berechnungen einfach auszudrücken und ihre Nutzung vermeidet das Speichern von Zwischenergebnissen. Dies reduziert den Speicherbedarf und die Datenbewegung und ermöglicht es, SPNs mit Millionen von Parametern auf größeren Datensätzen zu trainieren, was bisher unmöglich war.

4 Schlussfolgerungen

In dieser Dissertation haben wir Algorithmen entwickelt, um automatisch flexible und dennoch leistungsfähige probabilistische Verteilungen aus Daten zu erhalten und dabei die Anforderungen an die Nutzer*innen zu reduzieren. Wir leisten damit einen Beitrag zur Idee eines "Automatisierten Statistikers". Wir haben gezeigt, wie diese Modelle verwendet werden können, um Muster zu finden. Ein essentieller Vorteil zu bisherigen Modellen ist hierbei die Fähigkeit der Modelle Daten zu erkennen, für die sie nicht trainiert wurden. Unsere Beiträge sind über die Open-Source SPFlow-Software-Bibliothek [Mo19] leicht zugänglich. Die Flexibilität und die Bedeutung unserer Beiträge zeigen sich auch darin, dass sie substanzielle Verbesserungen in einem anderen Feld der Informatik, dem Bereich sehr großer Datenbanken (VLDB), ermöglichen [Hi20].

Literatur

[Hi20] Hilprecht, B.; Schmidt, A.; Kulessa, M.; Molina, A.; Kersting, K.; Binnig, C.: DeepDB: Learn from Data, not from Queries! International Conference on Very Large Data Bases (VLDB) 13/7, S. 992–1005, 2020.

[MNK17] Molina, A.; Natarajan, S.; Kersting, K.: Poisson Sum-Product Networks: A Deep Architecture for Tractable Multivariate Poisson Distributions. In: Proceedings of the 31st Conference on Artificial Intelligence (AAAI), San Francisco, USA. S. 2357–2363, 2017.

[Mo18] Molina, A.; Vergari, A.; Di Mauro, N.; Natarajan, S.; Esposito, F.; Kersting, K.: Mixed Sum-Product Networks: A Deep Architecture for Hybrid Domains. In: Proceedings of the 32nd Conference on Artificial Intelligence (AAAI), New Orleans, USA. S. 3828–3835, 2018.

[Mo19] Molina, A.; Vergari, A.; Stelzner, K.; Peharz, R.; Subramani, P.; Di Mauro, N.; Poupart, P.; Kersting, K.: SPFlow: An Easy and Extensible Library for Deep Probabilistic Learning Using Sum-Product Networks. arXiv:1901.03704/, 2019.

[Mo21] Molina, A.: Deep Networks That Know When They Don't Know, Diss., Darmstadt: Technische Universität, 2021.

[PD11] Poon, H.; Domingos, P. M.: Sum-Product Networks: A New Deep Architecture. In: Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI), Barcelona, Spain. S. 689–690, 2011.

[Pe14] Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Elsevier, 2014.

[Pe19] Peharz, R.; Vergari, A.; Stelzner, K.; Molina, A.; Trapp, M.; Shao, X.; Kersting, K.; Ghahramani, Z.: Random Sum-Product Networks: A Simple and Effective Approach to Probabilistic Deep Learning. In: Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence (UAI), Tel Aviv, Israel. S. 334–344, 2019.

[Pe20] Peharz, R.; Lang, S.; Vergari, A.; Stelzner, K.; Molina, A.; Trapp, M.; Broeck, G. V. d.; Kersting, K.; Ghahramani, Z.: Einsum Networks: Fast and Scalable Learning of Tractable Probabilistic Circuits. In: Proceedings of the 37th International Conference on Machine Learning (ICML). S. 7563–7574, 2020.

[Sh20] Shao, X.; Molina, A.; Vergari, A.; Stelzner, K.; Peharz, R.; Liebig, T.; Kersting, K.: Conditional sum-product networks: Imposing structure on deep probabilistic architectures. In: International Conference on Probabilistic Graphical Models. S. 401–412, 2020.

[Ve19] Vergari, A.; Molina, A.; Peharz, R.; Ghahramani, Z.; Kersting, K.; Valera, I.: Automatic Bayesian density analysis. In: Proceedings of the 33rd Conference on Artificial Intelligence (AAAI), Honolulu, USA. S. 5207–5215, 2019.

Alejandro Molina Ramirez wurde am 1983 in Medellin (Kolumbien) geboren. Nach seinem Bachelorstudium im Fach Informatik an der Universidad EAFIT in Medellin, erhielt er 2012 seinen Masterabschluss von der Albert-Ludwigs-Universität Freiburg. Als wissenschaftlicher Mitarbeiter von Prof. Kristian Kersting absolvierte er 2021 seine Promotion an der Technischen Universität Darmstadt. Er hat auf zahlreichen Top-Tier Konferenzen und in Fachzeitschriften veröffentlicht, darunter JMLR, VLDB, AAAI, ICLR, ICML und UAI. Zu seinen Forschungsinteressen gehören Deep Learning, Probabilistische Schaltungen und Maschinelles Lernen mit Big Data. Nachdem er bei Aleph Alpha an großen Sprachmodellen (GPT-3) gearbeitet hat, wechselte er zu Amazon.com als Machine Learning Scientist, wo er sich auf Kausalitätsmodelle konzentriert.

Unüberwachte Quantifizierung der Konsistenz von Entitäten zwischen Bild und Text in Nachrichtenartikeln[1]

Eric Müller-Budack[2]

Abstract: Öffentliche Nachrichten sind ein wichtiger Bestandteil unseres täglichen Lebens und über das Web wird eine immer größer werdende Zahl von Artikeln verbreitet. Zur Berichterstattung werden in der Regel multimodale Repräsentationen z. B. in Form von Texten und Fotos eingesetzt. Die Fotos können dekorativ sein, zusätzliche Details abbilden, aber ebenso wie Text auch irreführende Informationen enthalten. Die Quantifizierung der intermodalen Konsistenz von Nachrichten kann Nutzer*innen bei der Exploration und Bewertung der Glaubwürdigkeit unterstützen und potenziell Hinweise zur Erkennung von Fake News geben, die in der heutigen Gesellschaft zunehmend an Bedeutung gewinnen. In dieser Dissertation wird erstmals ein unüberwachter Ansatz zur Quantifizierung der intermodalen Bild-Text-Konsistenz von Entitäten, die eine zentrale Rolle in den Nachrichten einnehmen, vorgeschlagen. Um die zunehmende Anzahl von Entitäten in den Medien bewältigen zu können, werden die intermodalen Relationen, im Gegensatz zu bisherigen Forschungsansätzen, explizit und ohne zuvor definierte Trainings- und Exemplarbilder quantifiziert. Zu diesem Zweck werden geeignete Computer-Vision-Ansätze zur Extraktion von Ereignissen, Orten, Zeitangaben und Personen aus Nachrichtenfotos vorgestellt. Das vorgeschlagene System und die individuellen Komponenten werden in umfangreichen Experimenten evaluiert und erzielen vielversprechende Ergebnisse.

1 Einleitung

Das Web und die sozialen Medien übernehmen im heutigen Informationszeitalter eine wichtige Rolle für die Vermittlung von Nachrichten und Informationen. In der Regel werden verschiedene Modalitäten im Sinne der Informationskodierung wie beispielsweise Fotos und Text verwendet, um Nachrichten effektiver zu vermitteln oder Aufmerksamkeit zu erzeugen. Kommunikations- und Sprachwissenschaften erforschen das komplexe Zusammenspiel zwischen Modalitäten seit Jahrzehnten und haben unter anderem untersucht, wie durch die Kombination der Modalitäten zusätzliche Informationen oder neue Bedeutungsebenen (engl.: *meaning multiplication*) entstehen können [Ba14]. Die Anzahl der Konzepte oder Entitäten (z. B. Personen, Orte und Ereignisse), die im Foto und im Text vorkommen, und deren Konsistenz sind ein wichtiger Aspekt für die Bewertung der Gesamtaussage und Bedeutung eines multimodalen (Nachrichten-)Artikels.

Automatisierte Ansätze zur Quantifizierung von Bild-Text-Beziehungen (siehe Abb. 1) können für zahlreiche Anwendungen eingesetzt werden. Sie ermöglichen beispielsweise

[1] Englischer Titel der Dissertation: "Unsupervised Quantification of Entity Consistency between Photos and Text in Real-World News"

[2] Leibniz Universität Hannover, Forschungszentrum L3S, Appelstraße 4, 30167 Hannover, Deutschland, eric.mueller@tib.eu

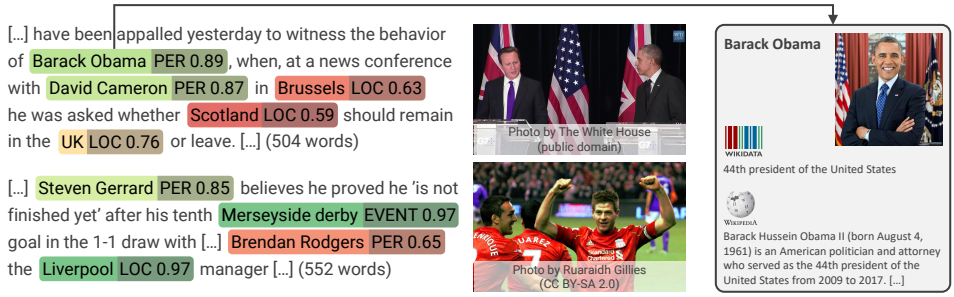


Abb. 1: **Links:** Beispielhafter multimodaler Nachrichtenartikel mit intermodalen Ähnlichkeiten für Orte (LOC), Personen (PER) und Ereignisse (EVENT) berechnet vom vorgeschlagenen Ansatz in dieser Dissertation. **Rechts:** Entitäteninformationen von *Wikipedia* und *Wikidata*. Eine Web Demo ist verfügbar unter: `https://labs.tib.eu/newsanalytics`

eine effektive und effiziente Exploration von Nachrichten, erleichtern die semantische Suche von Multimedia-Inhalten in (Web)-Archiven oder unterstützen Analyst:innen oder Fact-Checking-Seiten wie *PolitiFact*[3] und *Snopes*[4] bei der Evaluierung der Glaubwürdigkeit von Nachrichten. Es gab bereits einige Fälle in den (sozialen) Medien bei denen beispielsweise falsche Ortsangaben im Zusammenhang mit Nachrichtenvideos veröffentlicht wurden:

- Beispiel 1: "COVID-19: Old Video from Azerbaijan Shared as Lockdown in Spain"(Archivierter Weblink vom 12. April 2020: `https://bit.ly/3wSU5kZ`)
- Beispiel 2: "CBS admits crowded New York hospital was actually in Italy"(Archivierter Weblink vom 1. Februar 2021: `https://bit.ly/3wS5TE6`)

Bislang gibt es allerdings nur wenige Ansätze, die sich mit der Quantifizierung von Beziehungen zwischen Fotos und Text beschäftigen. Diese Ansätze berücksichtigen jedoch nicht explizit die intermodalen Beziehungen von Entitäten [Ot19; ZHK18], welche eine wichtige Rolle in Nachrichten darstellen, oder basieren auf überwachten multimodalen Deep-Learning-Techniken [Ja19; Sa18]. Diese überwachten Lernverfahren können ausschließlich die intermodalen Beziehungen von Entitäten bestimmen, die in (schwer akquirierbaren) annotierten Trainingsdaten enthalten sind und können somit mit der hohen Anzahl und der Diversität ständig wechselnder Entitäten und Themen in den Medien nicht umgehen.

Um diese Forschungslücke zu schließen, wird in der hier zusammengefassten Dissertation [Mü22] ein unüberwachter Ansatz zur Quantifizierung der intermodalen Konsistenz von Entitäten zwischen Fotos und Text in multimodalen Nachrichtenartikeln vorgestellt. Nachrichten berichten über weltweite Ereignisse und umfassen eine Vielzahl unterschiedlicher Entitäten wie z. B. Orte, Zeitangaben, Personen und die Ereignisse selbst. Die Extraktion von Entitäten aus Bild- und Textdaten ist deshalb essentiell zur Bestimmung

[3] `https://www.politifact.com/`
[4] `https://www.snopes.com/`

von intermodalen Relationen. Während Methoden der natürlichen Sprachverarbeitung zur Erkennung und Verlinkung von Entitäten im Allgemeinen gute Ergebnisse erzielen, haben Computer-Vision-Ansätze insbesondere bei Aufgaben, die ein hohes Maß an Szeneninterpretation benötigen (z. B. die Erkennung von Ereignissen und Orten), diverse Limitierungen. Abschnitt 2 gibt einen Überblick zu den in Kapitel 3 der Dissertation vorgeschlagenen Deep-Learning-Ansätzen zur Erkennung von Ereignissen, Orten, Zeitangaben und Personen in Fotos. Abschnitt 3 fasst das in Kapitel 4 der Dissertation vorgestellte unüberwachte System zur Quantifizierung von Bild-Text-Beziehungen in Nachrichten zusammen. Wie in Abb. 1 dargestellt, werden im Gegensatz zu bisherigen Verfahren neuartige Maße zur automatischen Schätzung der intermodalen Konsistenz für verschiedene Entitätstypen sowie den Gesamtkontext vorgeschlagen. Das System ist nicht auf vordefinierte Trainings- oder Exemplardaten angewiesen und kann daher besser mit der Vielzahl und Diversität von Entitäten und Themen in Nachrichten umgehen. In Abschnitt 4 werden die wichtigsten Forschungsergebnisse zusammengefasst und ein Ausblick auf zukünftige Arbeiten gegeben.

2 Informationsextration aus Fotos

2.1 Ereignisklassifikation in Fotos

In Nachrichten wird über Ereignisse aus verschiedenen Bereichen wie Kultur, Politik oder Sport berichtet, die für ein Zielpublikum von Bedeutung sind. Die bisher heterogenste Datensammlung *WIDER* [Xi15] für die Erkennung von Ereignissen unterscheidet zwischen 61 Ereigniskategorien. Allerdings deckt sie (wie andere Datensammlungen auch) viele wichtige Ereignistypen für Nachrichten, wie Epidemien oder Arten von Naturkatastrophen, nicht ab und beinhaltet lediglich rund 50.000 Fotos, davon circa 25.000 zum Training. Aufgrund des Mangels an Datensammlungen mit zahlreichen Bildern, beschränken sich verwandte Arbeiten vorwiegend auf Ensembles vortrainierter Modelle für ähnliche Aufgabenstellungen wie der Erkennung von Objekten und Szenen [Ah17; Ah18], sowie auf die Integration von Deskriptoren für lokale Bildregionen [Xi15].

Wie in Abb. 2 gezeigt, haben wir auf Basis von 550.994 realen Ereignissen über Relationen im *Wikidata* Wissensgraphen eine Ereignisonotologie (*VisE-O*) zur Unterscheidung von 148 Ereignistypen vorgestellt [Mü21]. Die dazugehörige Datensammlung (*VisE-D*) mit 570.540 Fotos, welche automatisch im Web gesammelt wurden, erlaubt das Training von Deep-Learning-Modellen und umfasst die bisher größte Menge an nachrichtenrelevanten Ereignistypen. Außerdem werden Ontologie-gestützte Deep-Learning-Modelle auf der Grundlage neuartiger Gewichtungsschemata und Verlustfunktionen vorgestellt, die die Ereignisbeziehungen aus den Informationen strukturierter Wissensgraphen nutzen. Auf diese Weise erhält das neuronale Netz zusätzliche Kontextinformationen, um die grundlegenden Ähnlichkeiten und Unterschiede verschiedener Ereignistypen (darunter Sportarten, soziale und kulturelle Ereignisse, Naturkatastrophen und Gesundheitskrisen) zu verstehen und daraus zu lernen.

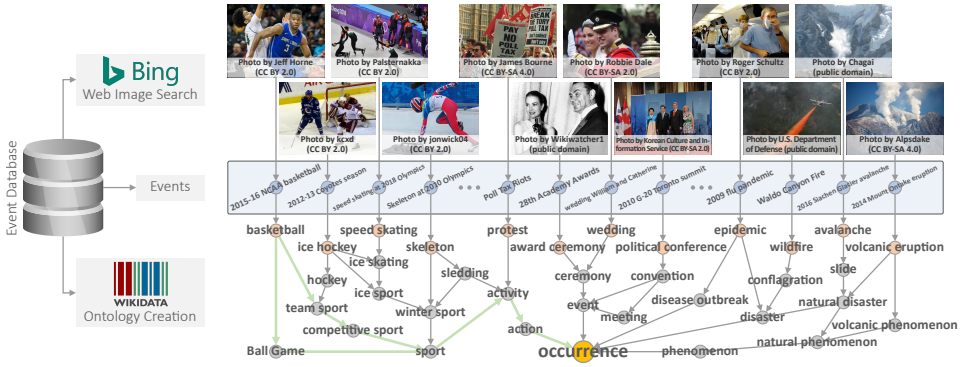


Abb. 2: Exemplarische Teilmenge der Eventontologie (*VisE-O*), sowie Fotos aus der Datensammlung (*VisE-D*). Die zu klassifizierenden Ereignistypen (orange) und anderen Knoten (grau) werden über Relationen des *Wikidata* Wissensgraphen zu einem Set an konkreten Ereignissen (blau) extrahiert.

Methode	**VisE-Bing**		**VisE-Wiki**		**WIDER [Xi15]**		**SocEID [Ah17]**		**RED [Ah17]**	
	Top-1	***JSC***	***Top-1***	***JSC***	***Top-1***	***JSC***	***Top-1***	***JSC***	***Top-1***	***JSC***
AlexNet [Xi15]	—	—	—	—	38,5	—	—	—	—	—
Event conc. [Ah17]	—	—	—	—	—	—	85,4	—	77,6	—
ResNet-152 [Ah18]	—	—	—	—	48,0	—	—	—	—	—
C	77,4	84,7	61,7	72,7	45,6	56,9	91,2	92,7	76,1	82,1
CO_{γ}^{cos}	81,9	87,9	**63,5**	**74,1**	**49,7**	**60,3**	**91,5**	**92,9**	**80,9**	**85,4**

Tab. 1: *Top-1* Genauigkeit und Jaccard-Koeffizient (*JSC*) der Ereignisklassifikation (Werte sind mit 100 multipliziert) auf verschiedenen Testdatensätzen. Notationen nach Abschnitt 3.1.4 der Dissertation.

Die experimentellen Ergebnisse in Tab. 1 für mehrere Testmengen, einschließlich zweier neuartiger Benchmarks (*VisE-Bing* und *VisE-Wiki*), zeigen die Überlegenheit des Ontologie-basierten Ansatzes (Notation: CO_{γ}^{cos}) gegenüber einer Klassifikationsbaseline (Notation: C), die die strukturierten Ontologie-Informationen nicht nutzt, sowie gegenüber vergleichbaren State-of-the-Art-Verfahren [Ah17; Ah18; Xi15]. Jedoch können mit dem vorgeschlagenen Ansatz nur Ereignistypen und keine konkreten Ereignisse klassifiziert werden, woraus sich je nach Applikation Einschränkungen ergeben können (siehe Abschnitt 4). Die Ontologie bietet aber eine gute Grundlage für zukünftige Arbeiten zur Ereignisidentifikation.

2.2 Schätzung des Aufnahmeortes von Fotos

Nachrichten beziehen sich in der Regel auch auf den Ort eines Ereignisses. Diese Ortsangaben können relativ unspezifisch und grob sein (Länder, Regionen, etc.), aber auch konkrete Angaben zu städtischen (Städte, Straßen, Gebäude, etc.) bis hin zu natürlichen Umgebungen (Gebirge, Meere, Wälder etc.) beinhalten. Deshalb werden zur Quantifizierung intermodaler

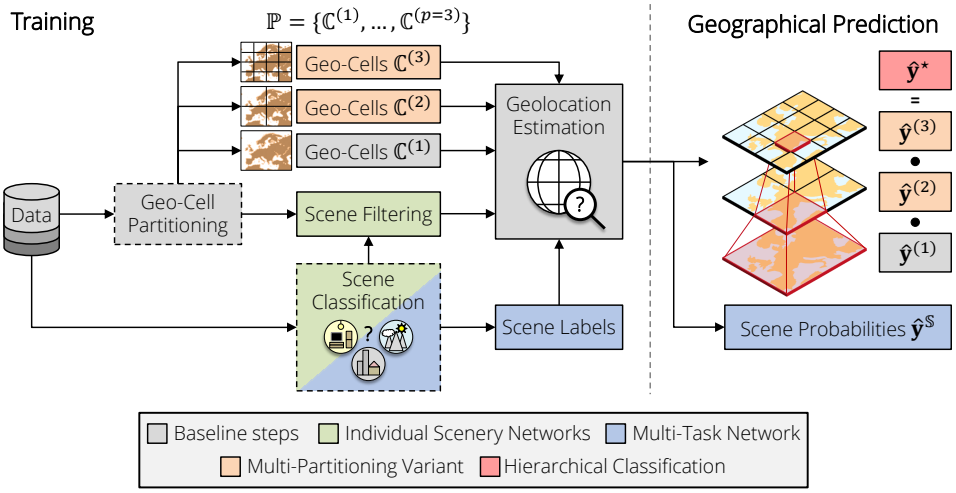


Abb. 3: Aufbau des vorgeschlagenen Verfahrens zur Schätzung des Aufnahmeortes von Bildern. Grau: Grundlegende Schritte, die in jedem Netzwerk enthalten sind. Zusätzliche Schritte für verschiedene Netzwerkkonfigurationen sind in verschiedenen Farben dargestellt. Die Schritte in den gestrichelten Rechtecken werden auf alle Bilder angewendet, bevor der Trainingsprozess stattfindet.

Relationen geografische Bildinformationen auf globaler Ebene, ohne Beschränkung auf bestimmte Umgebungen, benötigt.

Bisherige Ansätze [Se18; VJH17; WKP16] bestimmen die Bildposition über die Klassifikation geografischer Sektoren. Dazu wird die Erde in Sektoren mit ähnlich vielen Bildern aufgeteilt um Verzerrungen durch eine unausgeglichene geografische Verteilung von Fotos zu vermeiden. Allerdings bringt dies ein Trade-off-Problem mit sich. Zwar kann eine feinere Einteilung der Erde zu genaueren Vorhersagen führen, aber es sind auch weniger Fotos für jeden Sektor verfügbar. Dies kann zu einer Überanpassung des Modelles führen. Zudem stellen, selbst für moderne Deep-Learning-Architekturen, die enormen Variationen der Fotos aufgrund verschiedener Umgebungsbedingungen (z. B. Stadt-, Natur- und Innenaufnahmen), die spezifische Merkmale zur Unterscheidung erfordern, eine Herausforderung dar.

Für die oben genannten Probleme haben wir einen neuen Deep-Learning-Ansatz vorgeschlagen [MPE18]. Hierbei wird das neuronale Netz gleichzeitig mit mehreren Partitionierungen unterschiedlicher Auflösung trainiert (siehe Abb. 3, orange), um sowohl lokale und globale geografische Informationen zu erlernen und das angesprochene Trade-off-Problem zu adressieren. Die Ausgaben des Netzes werden für die verschiedenen Granularitäten kombiniert, um die Vorhersage auf der feinsten Partitionierungsebene zu verbessern (siehe Abb. 3, rot). Darüber hinaus schlagen wir zwei Strategien zur Einbeziehung von Szeneninformationen vor: (a) Individuelle Szenennetze (*ISNs*; Abb. 3, grün), die separat mit Stadt-, Natur-, und Innenaufnahmen trainiert werden, und (b) ein Multi-Task-Netz (*MTN*; Abb. 3, blau), das simultan für die Geolokalisierung und die Szenenklassifikation trainiert wird.

Methode	I_T	I_R	Straße 1 km	Stadt 25 km	Region 200 km	Land 750 km	Kontinent 2.500 km
Im2GPS, kNN, $\sigma = 4$ [VJH17]	6,0M	6,0M	7,2 %	19,4 %	26,9 %	38,9 %	55,9 %
PlaNet (reprod. by [Se18])	30,3M	—	8,5 %	24,8 %	34,3 %	48,4 %	64,6 %
CPlaNet (best) [Se18]	30,3M	—	10,2 %	26,5 %	34,6 %	48,6 %	64,6 %
base (L, m)	4,7M	—	8,3 %	24,9 %	34,0 %	48,8 %	65,8 %
ISNs $(M, f*, \mathbb{S}_3)$	4,7M	—	**10,5 %**	**28,0 %**	**36,6 %**	**49,7 %**	**66,0 %**

Tab. 2: Anzahl an Trainings (I_T)- und Retrievalfotos (I_R), sowie Anteil an Testfotos lokalisiert mit verschiedenen Genauigkeiten auf Basis der geodätischen Distanz [%] auf dem *Im2GPS3k* Testdatensatz (2.997 Testfotos). Notationen nach Abschnitt 3.2.4 der Dissertation.

Experimentelle Ergebnisse für zwei verschiedenen Benchmarks (Ergebnisse für Benchmark *Im2GPS3k* in Tab. 2) haben gezeigt, dass der vorgeschlagene Ansatz basierend auf multiplen Erdpartitionen und individuellen Szenennetzen (*ISNs*) die besten Ergebnisse liefert und den Stand der Forschung übertrifft. Dabei verwendet der Ansatz keine zusätzlichen Retrieval-Methoden (*Im2GPS* [VJH17]) und eine deutlich geringere Anzahl von Trainingsbildern im Vergleich zu *PlaNet* und *CPlaNet* [Se18]. Ein Webdemonstrator[5] des Ansatzes wurde auf Ausstellungen (z. B. MS Wissenschaft, Deutsches Museum Bonn) präsentiert und in einem Artikel des Computermagazins *c't*[6] beschrieben.

2.3 Schätzung des Aufnahmejahres von Fotos

Bislang existierten noch keine Arbeiten zur uneingeschränkten (beliebige Bildmotive, Schwarz-Weiß- und Farbfotografie) Schätzung des Aufnahmejahres von (historischen) Fotos. Daher haben wir eine erste Datensammlung mit mehr als einer Million Bildern aus den Jahren 1930 bis 1999 erstellt [MSE17]. Diese Datensammlung dient zum Training von zwei Deep-Learning-Verfahren, welche die Aufgabe als Regressions- und Klassifikationsproblem lösen. Der Klassifikationsansatz teilt die Bilder in Perioden von fünf Jahren auf und interpoliert die vorhergesagten Wahrscheinlichkeiten zur Schätzung eines Aufnahmejahres.

Beide Ansätze übertreffen mit einem durchschnittlichen Fehler von 7,3 Jahren (Klassifikation) und 7,5 Jahren (Regression) auf 1.120 Testfotos die Annotationen menschlicher Proband*innen (Fehler: 10,9 Jahre). Allerdings sind die Anwendungsmöglichkeiten des Verfahrens durch die Beschränkung auf Aufnahmejahre bis 1999 und die Höhe des durchschnittlichen Fehlers limitiert. Zeitgenössische Fotos variieren je nach den zugrundeliegenden geografischen, wirtschaftlichen und kulturellen Merkmalen des Aufnahmestandortes und erfordern die Modellierung dieser Merkmale. Deshalb stellt die Identifikation konkreter Ereignisse (siehe Abschnitt 2.1), die üblicherweise mit zeitlichen und geografischen Informationen verknüpft sind, einen vielversprechenderen Ansatz dar.

[5] Web Demo: `https://labs.tib.eu/geoestimation/`
[6] c't Artikel: `www.heise.de/select/ct/2019/5/1551091142351937`

2.4 Personenerkennung in Nachrichtenbildern des Internetarchivs

Im Gegensatz zu den vorherigen Computer-Vision-Teilgebieten ist die Identifikation von Personen seit Jahrzehnten gut erforscht. Deshalb wurde in der Dissertation eine konkrete Anwendung zur automatischen Erkennung relevanter Personen und deren Relationen in Nachrichtenbildern des Internetarchivs[7] untersucht.

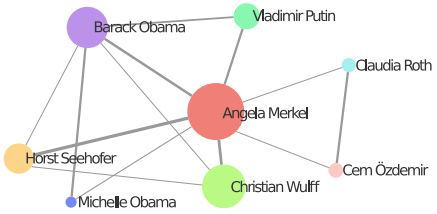


Abb. 4: Frequenz des Auftretens von Entitäten (Knotenradius) und des gemeinsamen Auftretens (Kantenstärke) extrahiert aus Nachrichtenbildern des Internetarchivs (Stand 2013)

Hierzu haben wir einen Multimedia-Retrieval-Ansatz vorgestellt, der automatisch ein Lexikon mit den relevantesten Personen einer Domäne (z. B. Politik oder Unterhaltung) für einen Zeitraum erstellt und diese mit Beispielbildern aus dem Web identifiziert [MPE18]. Da die Ergebnisse der Bildersuche auch irrelevante Fotos enthalten, auf denen mehrere oder unterschiedliche Personen abgebildet sind, wird ein zusätzlicher Filterungsschritt auf Basis eines agglomerativen Clusterings angewendet. Schließlich werden Gesichtsmerkmale durch einen Deep-Learning-Ansatz extrahiert und mit denen der Nachrichtenbilder verglichen, um Personen in den Bilddaten des Internetarchivs zu identifizieren. Eine Fallstudie (Abb. 4) und Ergebnisse auf dem Benchmark *Labeled Faces in the Wild*[8] (98 % Verifikationsgenauigkeit) hat gezeigt, dass Personen in Nachrichtenbildern zuverlässig identifiziert werden können.

3 Quantifizierung der intermodalen Konsistenz von Nachrichten

Zur Quantifizierung der intermodalen Konsistenz in Nachrichten haben wir das erste unüberwachte System (Abb. 5) vorgeschlagen [Mü21] . Im Gegensatz zu bisherigen Verfahren [Ja19; Ot19; Sa18; ZHK18] ist es in der Lage, differenzierte Bild-Text-Beziehungen für eine hohe Vielzahl und Diversität von Entitäten in den Nachrichten zu bestimmen. Zur Detektion und Verlinkung von Entitäten im Text zu *Wikidata* wird spaCy[9] in Kombination mit *Wikifier* [BLG17] eingesetzt. Anschließend werden Beispielbilder für die Entitäten automatisch mithilfe von Bildsuchmaschinen (*Google*, *Bing*) und *Wikidata* akquiriert. Wie in Abschnitt 2.4 wird ein agglomeratives Clustering angewendet, um die heruntergeladenen Bilder für Personen zu filtern. Diese Bilder dienen als visuelle Evidenz für die Überprüfung der intermodalen Konsistenz der detektierten Entitäten zum zugehörigen Nachrichtenbild. Zu diesem Zweck werden die vorgeschlagenen Computer-Vision-Ansätze aus Abschnitt 2 als verallgemeinerte Merkmalsextraktoren verwendet. Schließlich werden neuartige Maße für verschiedene Entitätstypen (Personen, Orte, Ereignisse), sowie für den allgemeineren

[7] Webseite des Internetarchives: `https://archive.org/`
[8] Webseite von Labeled Faces in the Wild: http://vis-www.cs.umass.edu/lfw/
[9] Webseite von spaCy: `https://spacy.io/`

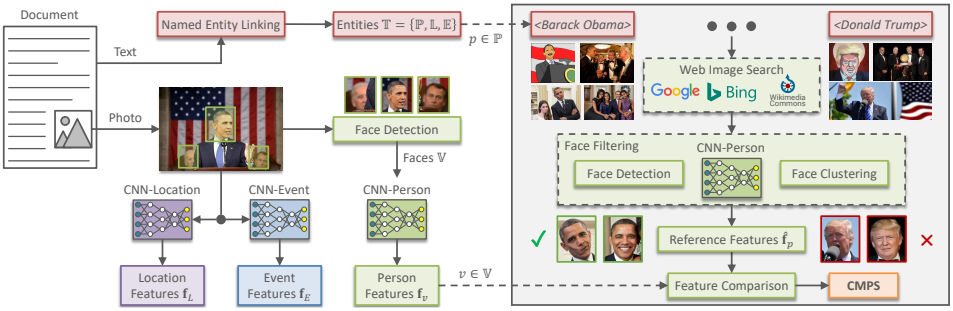


Abb. 5: Übersicht des vorgeschlagenen Systems zur Quantifizierung der intermodalen Konsistenz von Entitäten. **Links:** Extraktion von Entitäten aus dem Text, sowie von Bildmerkmalen für Personen (grün), Orte (lila) und Ereignisse (blau). **Rechts:** Quantifizierung der intermodalen Personenkonsistenz anhand von Webbildern. Eine ähnliche Pipeline (ohne Filterung) wird für Orte und Ereignisse verwendet.

Nachrichtenkontext eingeführt, um die intermodale Konsistenz zu quantifizieren. Zwei Beispielresultate und der Link zur Web Demo sind Abb. 1 zu entnehmen.

Experimentelle Ergebnisse (Abschnitt 4.5 der Dissertation) für die Verifikation und das Retrieval von Nachrichten mit geringer (potenzielle Fehlinformation) bzw. hoher multimodaler Konsistenz auf zwei Testdatensammlungen (englisch und deutsch) haben das Potenzial des Ansatzes gezeigt. Insbesondere die Ergebnisse für Personen, feingranulare Orte (Städte, Gebäude, Straßen, etc.) und Ereignistypen sind sehr vielversprechend. Allerdings ist die Güte des Systems abhängig von den Ergebnissen der Bildsuchmaschinen. Oftmals werden z. B. für grobe Ortsangaben (Länder, Gebirge, etc.) bzw. für mehrdeutige oder weniger populäre Entitäten Bilder heruntergeladen, die in Bezug auf den Nachrichtenkontext und zur Bestimmung intermodaler Relationen irrelevant sind. Zudem würde das System von einem Ansatz profitieren, dass zwischen konkreten Ereignissen unterscheiden kann (Abschnitt 2.1).

4 Zusammenfassung

In dieser Dissertation [Mü22] wurde erstmals ein unüberwachter Ansatz zur automatischen Quantifizierung der intermodalen Konsistenz von Nachrichten vorgeschlagen und erforscht. In diesem Zusammenhang wurden neuartige Deep-Learning-Ansätze zur Erkennung von Ereignissen, Orten, Zeitangaben und Personen aus Fotos vorgestellt, welche vergleichbare Referenz- und State-of-the-Art-Verfahren auf Benchmarkdaten schlagen. Im Gegensatz zu verwandten Arbeiten können differenzierte intermodale Relationen von Entitäten ohne zusätzliche Trainings- oder vordefinierte Beispielbilder bestimmt werden. Die vorgeschlagenen Methoden können z. B. Analyst*innen eine effiziente Exploration von Nachrichtenkorpora ermöglichen und bei der Untersuchung der Glaubwürdigkeit von Nachrichten unterstützen.

In Zukunft sollen weitere Entitätstypen wie Organisationen in das System integriert werden. Ein weiterer wichtiger Aspekt ist die Bestimmung der multimodalen Konsistenz konkreter

Nachrichtenereignisse, da diese sowohl zeitliche als auch geografische Informationen beinhalten können. Zudem können durch verbesserte Suchanfragen, beispielsweise auf Grundlage des konkreten Nachrichtenkontextes, passendere Beispielbilder gefunden werden, um die Güte des Systems weiter zu steigern.

Literatur

[Ah17] Ahsan, U.; Sun, C.; Hays, J.; Essa, I. A.: Complex Event Recognition from Images with Few Training Examples. In: IEEE Winter Conference on Applications of Computer Vision. IEEE Computer Society, S. 669–678, 2017.

[Ah18] Ahmad, K.; Mekhalfi, M. L.; Conci, N.; Melgani, F.; De Natale, F. G. B.: Ensemble of Deep Models for Event Recognition. ACM Trans. Multim. Comput. Commun. Appl. 14/2, 51:1–51:20, 2018.

[Ba14] Bateman, J.: Text and image: A critical introduction to the visual/verbal divide. Routledge, 2014.

[BLG17] Brank, J.; Leban, G.; Grobelnik, M.: Annotating Documents with Relevant Wikipedia Concepts. In: Slovenian Conference on Data Mining and Data Warehouses. 2017.

[Ja19] Jaiswal, A.; Wu, Y.; AbdAlmageed, W.; Masi, I.; Natarajan, P.: AIRD: Adversarial Learning Framework for Image Repurposing Detection. In: IEEE Conference on Computer Vision and Pattern Recognition. Computer Vision Foundation / IEEE, S. 11330–11339, 2019.

[MPE18] Müller-Budack, E.; Pustu-Iren, K.; Ewerth, R.: Geolocation Estimation of Photos Using a Hierarchical Model and Scene Classification. In: European Conference on Computer Vision. Bd. 11216, Springer, S. 575–592, 2018.

[MSE17] Müller, E.; Springstein, M.; Ewerth, R.: "When Was This Picture Taken? Image Date Estimation in the Wild. In: European Conference on Information Retrieval. Bd. 10193, S. 619–625, 2017.

[Mü18] Müller-Budack, E.; Pustu-Iren, K.; Diering, S.; Ewerth, R.: Finding Person Relations in Image Data of News Collections in the Internet Archive. In: International Conference on Theory and Practice of Digital Libraries. Bd. 11057, Springer, S. 229–240, 2018.

[Mü20] Müller-Budack, E.; Theiner, J.; Diering, S.; Idahl, M.; Ewerth, R.: Multimodal Analytics for Real-world News using Measures of Cross-modal Entity Consistency. In: International Conference on Multimedia Retrieval. ACM, S. 16–25, 2020.

[Mü21] Müller-Budack, E.; Springstein, M.; Hakimov, S.; Mrutzek, K.; Ewerth, R.: Ontology-driven Event Type Classification in Images. In: IEEE Winter Conference on Applications of Computer Vision. IEEE, S. 2927–2937, 2021.

[Mü22] Müller-Budack, E.: Unsupervised quantification of entity consistency between photos and text in real-world news, Diss., Gottfried Wilhelm Leibniz Universität Hannover, 2022.

[Ot19] Otto, C.; Springstein, M.; Anand, A.; Ewerth, R.: Understanding, Categorizing and Predicting Semantic Image-Text Relations. In: International Conference on Multimedia Retrieval. ACM, S. 168–176, 2019.

[Sa18] Sabir, E.; AbdAlmageed, W.; Wu, Y.; Natarajan, P.: Deep Multimodal Image-Repurposing Detection. In: ACM Multimedia. ACM, S. 1337–1345, 2018.

[Se18] Seo, P. H.; Weyand, T.; Sim, J.; Han, B.: CPlaNet: Enhancing Image Geolocalization by Combinatorial Partitioning of Maps. In: European Conference on Computer Vision. Bd. 11214, Springer, S. 544–560, 2018.

[VJH17] Vo, N. N.; Jacobs, N.; Hays, J.: Revisiting IM2GPS in the Deep Learning Era. In: IEEE International Conference on Computer Vision. IEEE Computer Society, S. 2640–2649, 2017.

[WKP16] Weyand, T.; Kostrikov, I.; Philbin, J.: PlaNet - Photo Geolocation with Convolutional Neural Networks. In: European Conference on Computer Vision. Bd. 9912, Springer, S. 37–55, 2016.

[Xi15] Xiong, Y.; Zhu, K.; Lin, D.; Tang, X.: Recognize complex events from static images by fusing deep channels. In: IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, S. 1600–1609, 2015.

[ZHK18] Zhang, M.; Hwa, R.; Kovashka, A.: Equal But Not The Same: Understanding the Implicit Relationship Between Persuasive Images and Text. In: British Machine Vision Conference. BMVA Press, S. 8, 2018.

Eric Müller-Budack wurde am 10. März 1989 in Jena (Deutschland) geboren. Er erlangte die Abschlüsse Bachelor of Engineering (Jahr 2012) und Master of Engineering (Jahr 2014) an der Ernst-Abbe-Hochschule Jena. Seine Masterarbeit wurde mit der Note 1,0 bewertet und mit dem Thüringer VDI-Preis für die beste Studienabschlussarbeit ausgezeichnet. Von 2014 bis 2016 war er als wissenschaftlicher Mitarbeiter an der Ernst-Abbe-Hochschule tätig. Seit 2016 arbeitet er als wissenschaftlicher Mitarbeiter in der Forschungsgruppe Visual Analytics der Technischen Informationsbibliothek (TIB), sowie am Forschungszentrum L3S der Leibniz Universität Hannover. Seine Forschungsschwerpunkte liegen in den Bereichen Computer Vision, Deep Learning, sowie Multimedia Information Retrieval. Während seiner Promotion hat er an mehreren Drittmittelprojekten (DFG, BMWi, BMBF) und 22 Publikationen (davon neun als Erstautor) mitgewirkt. Zudem erhielt er den Honorable Mention Award bei der TPDL 2018 [Mü18] und den Best Paper Award bei der ICMR 2020 [Mü20]. Am 10. Dezember 2021 hat er seine Promotion an der Leibniz Universität Hannover mit Auszeichnung abgeschlossen.

Generische Verkettung maschineller Ansätze der Bilderkennung durch Wissenstransfer in verteilten Systemen

Am Beispiel der Aufgabengebiete INS und ACTEv der Evaluationskampagne TRECVid

Christian Roschke[1]

Abstract: Technologischer Fortschritt im Bereich multimedialer Sensorik und zugehörigen Methoden zur Datenaufzeichnung, Datenhaltung und -verarbeitung führt im Big Data-Umfeld zu immensen Datenbeständen in Mediatheken und Wissensmanagementsystemen. Dabei Zugrundliegende State of the Art-Verarbeitungsalgorithmen werden oftmals problemorientiert entwickelt, wobei sich aufgrund der enormen Datenmengen nur bedingt zuverlässig Rückschlüsse auf Güte und Anwendbarkeit ziehen lassen. So gestaltet sich auch die intellektuelle Erschließung von großen Korpora schwierig, da die Datenmenge für valide Aussagen nahezu vollumfänglich semi-intellektuell zu prüfen wäre, was spezifisches Fachwissen aus der zugrundeliegenden Datendomäne ebenso voraussetzt wie zugehöriges Verständnis für Datenhandling und Klassifikationsprozesse. Ferner gehen damit gesonderte Anforderungen an Hard- und Software einher, welche in der Regel suboptimal skalieren, da diese zumeist auf Multi-Kern-Rechnern entwickelt und ausgeführt werden, ohne dabei eine notwendige Verteilung vorzusehen. Folglich fehlen Mechanismen, um die Übertragbarkeit der Verfahren auf andere Anwendungsdomänen zu gewährleisten. Die Arbeit [Ro21] nimmt sich diesen Herausforderungen an und fokussiert auf die Konzeptionierung und Entwicklung einer verteilten holistischen Infrastruktur, die die automatisierte Verarbeitung multimedialer Daten im Sinne der Merkmalsextraktion, Datenfusion und Metadatensuche innerhalb eines homogenen Systems ermöglicht.

1 Einführung

Der technologische Fortschritt im Bereich multimedialer Sensorik und zugehörigen Methoden zur Datenaufzeichnung, Datenhaltung und -verarbeitung führt im *Big Data*-Umfeld u.a. zu immensen Datenbeständen in Mediatheken, Medienarchiven und Wissensmanagementsystemen. Zugrundliegende *State of the Art*-Verarbeitungsalgorithmen werden oftmals problemorientiert entwickelt. Aufgrund der enormen Datenmengen lassen sich nur bedingt zuverlässig Rückschlüsse auf Güte und Anwendbarkeit ziehen. So gestaltet sich auch die intellektuelle Erschließung von großen Korpora schwierig, da die Datenmenge für valide Aussagen nahezu vollumfänglich zumindest semi-intellektuell zu prüfen wären, was spezifisches Fachwissen aus der zugrundeliegenden Datendomäne ebenso voraussetzt wie zugehöriges Verständnis für Datenhandling und Klassifikationsprozesse in der Informations- und Kommunikationstechnik. Ferner gehen damit gesonderte Anforderungen an Hard- und Software einher, welche sich in der Regel schlecht skalieren, da die-

[1] christian.roschke@hs-mittweida.de

se zumeist auf Multi-Kern-Rechnern entwickelt und ausgeführt werden ohne dabei eine notwendige Verteilung vorzusehen. Demzufolge fehlen auch Mechanismen, um die Übertragbarkeit der Verfahren auf andere Anwendungsdomänen mit geringen Aufwänden zu gewährleisten.

Der Fokus der Arbeit liegt in der Konzeptionierung und Entwicklung einer verteilten holistischen Infrastruktur, die die automatisierte Verarbeitung multimedialer Daten im Sinne der Merkmalsextraktion, Metadatenanreicherung, Metadatenfusion und Metadatensuche mit Hilfe von modularen Komponenten innerhalb eines homogenen aber zugleich verteilten Systems ermöglicht. Dabei sind Ansätze aus den Domänen des *Maschinellen Lernens*, der *Verteilten Systeme*, des *Datenmanagements* und der *Virtualisierung* zielführend miteinander zu verknüpfen, um auf große Datenmengen angewendet, evaluiert und optimiert werden zu können.

2 Herausforderungen zur Optimierung des Wissenstransfers zwischen SMAs

Der Prozess des *Information Retrieval* besteht nach *Kürsten* [Kü12] und *Ritter* [Ri14] aus mehreren Phasen. Nach dem Upload von multimedialen Daten (Text, Bild oder Video) in ein Datenrepository folgt die Annotation ausgewählter Datensamples, um eine Extraktion von Metadaten mit Hilfe maschineller Lernverfahren und automatisierter Methoden zu realisieren, deren Leistungsfähigkeit sich mittels komponentenbasierter Evaluation und Optimierung steigern lässt.

Beide Autoren fokussieren in ihren Arbeiten auf die Optimierung von *Single Model Architekturen* (SMA) in jeweiligen Problemdomänen. Unter dem Begriff SMA werden in diesem Zusammenhang Architekturen verstanden, die hauptsächlich auf einem Modell basieren, dabei zwar multiple Ansätze des maschinellen Lernens anzuwenden vermögen, aber explizit für ein Problemfeld kreiert wurden und zumeist auf einem spezifischen Einzelrechner-System funktionieren. Der dabei untersuchte Wissenstransfer erfolgt in einer kontrollierten und fest definierten Umgebung innerhalb eines Modells. Das zu transferierende Wissen ist dabei als Sammlung von Metainformationen sowie Hyperparametern zu verstehen, wobei der Wissenstransfer ausschließlich innerhalb einer SMA erfolgt bzw. genutzt wird, um mithilfe von Ensemble-Ansätzen die Güte der Detektionsergebnisse zu verbessern.

Nach *Ritter* [Ri14, §§4,5] funktionieren diese Ansätze bei homogeneren Problemstellungen wie der strukturellen Videoanalyse und dem dortigen automatisierten Auffinden von Kameraeinstellungen und Übergängen (*Shot Detection*) oder bei der Lokalisation von Gesichtern oder Personen in Einzelbildern oder Videos. Im Gegensatz zu diesen homogeneren Problemfeldern umfassen die TRECVid Aufgabengebiete *Instance Search* (INS) und *Activities in Extended Video* (ACTEv) wesentlich komplexere, teilweise aus mehreren einfacheren Problemfeldern zusammengefasste Aufgabenstellungen. INS stellt die Suche nach unscharf vorgegebenen Objektklassen dar, wobei neben der Detektion und Erkennung auch die Identifizierung bestimmter Personen an definierten Orten im Fokus steht.

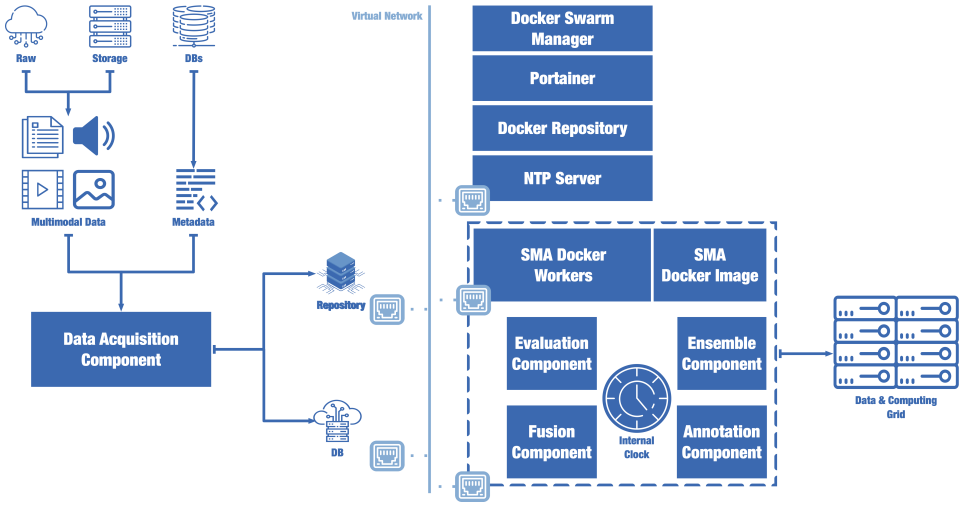


Abb. 1: Vereinfachte Darstellung der Gesamtarchitektur von EMSML

Dies bedingt eine Kombination von Objekterkennung, Ortserkennung und Gesichtserkennung. Die Suchprozesse sind darüber hinaus ad-hoc durchzuführen und beliebige Topics ohne anfragespezifisches Training zu verarbeiten. Bei ACTEv hingegen sind Entitäten und deren Aktivitäten innerhalb eines Datensatzes zu identifizieren. Dabei wird auf die Interaktion zwischen detektierten Objektklassen fokussiert und es stehen wenig *Ground Truth*-Daten zur Verfügung. Die Komplexität ist darüber hinaus dadurch gegeben, dass die Kombinationen von Objekterkennung, Tracking und Aktivitätserkennung essentiell sind, um die gestellten Suchanfragen behandeln zu können. Überdies werden die Datensätze vier bis acht Wochen und die Suchanfragen zwei Wochen vor der finalen Abgabe bereitgestellt, was eine zeitlich eingeschränkte Bearbeitungszeit bedingt.

Zur Bearbeitung derart komplexer und heterogener Problemfelder in adäquater Rechenzeit werden skalierbare und verteilte Ansätze benötigt, die in der Lage sind, multiple SMAs verteilt über diverse Rechnersysteme anzuwenden und einen Wissenstransfer zwischen diesen zu realisieren. Diesbezüglich sind Aspekte des *Datenhandling & Datenmanagement*, der *Steuerung & Virtualisierung*, der *Modellierung & Erstellung von Domänenwissen* sowie der *Annotation, Evaluation und Optimierung* zu beachten und in einer generischen sowie holistischen Infrastruktur abzubilden. Eine dafür geeignete Architektur stellt das in der Arbeit entwickelte und in Abbildung 1 vereinfacht dargestellte *Evaluations- und Managementsystem für Maschinelles Lernen* (EMSML) dar.

Im Rahmen von *Datenhandling und Datenmanagement* ist das Gesamtsystem mittels der *Data Acquisition Component* in der Lage, multimodale Daten sowie Metadaten aus unterschiedlichen Quellen zu importieren und für die persistente Speicherung aufzubereiten. Im Sinne der Vereinheitlichung heterogener Daten findet diesbezüglich eine Umbennung und Transformation in homogene Formate statt. Die Speicherung erfolgt letztendlich auf Dateiebene in Repositories und auf Metadatenebene in Datenbanken. Dabei kann das Wissen,

aus den in den jeweiligen Problemdomänen ursprünglich zur Verfügung gestellten Daten, gesammelt und durch zentrale sowie ausfallsichere Zugriffssysteme problemübergreifend transferiert werden.

Die *Steuerung und Virtualisierung* erfolgt mittels *virtualisiertem Netzwerk*, das den Zugriff auf die Infrastruktur ermöglicht. SMAs tauschen sich darüber aus und können zur Verarbeitung bereits gesammeltes Wissen verwenden. Über eine *Docker Registry* ist das System darüber hinaus in der Lage, Zustandsinformationen einzelner *Single Model Architekturen* in Form von *Docker Images* bereitzustellen und zu verteilen. Die einzelnen SMAs selbst werden als *Docker Container* virtualisiert, wobei eine Verteilbarkeit der Businesslogik auf diverse Rechenknoten stattfinden kann. Die Verteilung dieser Logik übernimmt der *Docker Swarm Manager*, der jeden *Docker Container* als zentrale *Broker-Instanz* verwaltet sowie den Wissenstransfer auf technischer Ebene unter Bereitstellung notwendiger Ressourcen ermöglicht. Mittels *Portainer* lässt sich die Wissensbasis darüber hinaus intellektuell verwalten und zusätzlich mit semantischen Informationen, Prozessabläufen etc. anreichern.

Die Modellierung & Erstellung von Domänenwissen erfolgt über den Einsatz von *Apache Spark*, wobei *DataFrames* zur Verarbeitung bereitgestellt werden und eine Abfrage über *Spark SQL* realisiert wird. Dabei lässt sich das übermittelte Wissen in Form von Metadaten während der Übertragung beliebig modellieren, filtern und problemspezifisch anpassen. Die Übermittlung selbst erfolgt in diesem Kontext batchweise oder in Echtzeit und realisiert somit eine Verarbeitung im *Big Data*-Umfeld. Diesbezüglich können parallele Anfragen mehrerer *Docker-Container* beliebig kombiniert oder ausgetauscht werden, wobei die Verwaltung über den *Swarm Manager* unter Einbezug einer mittels NTP synchronisierten Zeit koordiniert wird.

Die *Annotations*-Komponente innerhalb von EMSML stellt eine webbasierte Annotationsplattform für Audio- und Videoanalysealgorithmen zur Verfügung, wobei die Güte von Prozessketten und SMAs zusätzlich zu automatisch erzeugten Metriken intellektuell eingeschätzt werden kann. Darüber hinaus lässt sich damit zusätzliches semantisches Wissen hinzufügen, wobei Annotationen von Benutzern zeitgenau erfassbar sind. Die automatisierte *Evaluation* wird über Vergleichssysteme innerhalb von EMSML realisiert. Diesbezüglich findet eine Aufgliederung komplexer Probleme in weniger komplexe Teilprobleme statt. Dabei sind einzelne SMAs anzuwenden und mittels klassischer Metriken im Kontext der Detektionsgüte einschätzbar. Überdies ist dabei der Einsatz von Kreuzvalidierung zur *Optimierung* der Hyperparametereinstellungen innerhalb einzelner oder mehrerer SMAs realisierbar, was eine Identifizierung des besten Ansatzes für jeweils ein Teilproblem und durch geeignete Kombination eine möglichst optimale Lösung des Anwendungsproblems zulässt.

3 Wissenschaftliche Ergebnisse

Diese Arbeit ordnet Methoden und Strategien des State of the Art im Bereich des Maschinelles Lernen in der Mediendomäne unter Inklusion der klassischen Lernmethoden zzgl. Transfer-Lernen, Ensemble-Lernen und zugehöriger Leistungemaße ein und betrachtet aktuelle Technologien und Frameworks zur Detektion von Mustern in den Domänen zur Gesichts-, Aktivitäts- und Ortserkennung. Zudem werden diese ins Verhältnis zum aufkommenden Datenmanagement gesetzt und mit den Erfordernissen und Architekturmodellen verteiler Systeme und Virtualisierungslösungen verbunden.

Um den Anforderungen verteilter Systeme gerecht zu werden, wird das von Ritter [Ri14] abgeleitete Schema zur generischen Mustererkennung dabei in einem ersten Schritt um drei Strategien zur Datenfusion auf Pixel-Level, Feature-Level und Decision-Level geeignet erweitert. Zur Demonstration der Allgemeingültigkeit erfolgt eine Eingruppierung der einzelnen Prozessschritte von neuronalen Faltungsnetzen (CNN). Aus der intensiven Betrachtung der Fachliteratur werden insgesamt zehn Hauptkriterien abgeleitet, welche notwendig sind, um an multimedialen wissenschaftlichen Evaluationskampagnen unter Einsatz von verteilter Systemen in verschiedenen Problemdomänen teilnehmen zu können.

Der holistische Systementwurf der in der Arbeit erstellten Gesamtarchitektur des *Evaluations- und Managementsystems für Maschinelles Lernen* (EMSML) besteht aus elf Hauptkomponenten und 30 Anforderungen, welche über technische und qualitative Anforderungen hinaus Aspekte und Konzepte zum Datenhandling, zur Ausgestaltung der *Single Model Architekturen* und zum Wissenstransfer enthalten. Ein domänenübergreifender Workflow identifiziert sieben Basisschritte, um kompatible und verteilbare SMAs zu erstellen, die problemspezifisch entworfen wurden, deren (Teil-)Ergebnisse aber problemübergreifend zur Lösung komplexerer Problemklassen beitragen können.

Eine tiefgreifenden Analyse der Entwicklung der Problemdomäne *Instance Search* und des *Related Work* offenbart über die Historie der wissenschaftlichen Evaluationskampagne einen steigenden Schwierigkeitsgrad, der sich in zunehmend fordernden Kategorien und höherem Automatisierungsgrad widerspiegelt. Konkret sei hierbei darauf verwiesen, dass zu Beginn des Wettbewerbs eher einzelne Objekte aufzufinden waren, für die vage Umschreibungen wie *cuved plastic bottle of ketchup* und vier bis fünf Beispielbilder existierten. Im Verlauf der Zeit und der Entwicklung der zugrundeliegenden Technologien in den vergangenen für diese Arbeit relevanten Evaluationsperioden trat hingegen die Suche nach eher selten auftretenden Nebendarstellern mit unterschiedlichen Aktivitäten oder unterschiedlichen Orten in den Fokus. Im Vergleich zur traditionellen Objekterkennung, begründet sich die Schwierigkeit des Problems in vier wesentlichen Punkten: (a) Eine äußerst geringe Anzahl an Trainingsbeispielen. (b) Das zu Trainingszwecken verwendbare Videomaterial ist zwei Stunden lang und für einen Großteil der Suchanfragen nicht repräsentativ, da dies willkürlich ausgewählt wurde (Video 0 als erstes Video der gesamten Kollektion) und in der Regel weder das gesuchte Objekt noch dessen Ort oder Handlung enthält. (c) Der Testkorpus ist mit 462 Stunden Länge überproportional groß und darf in keinem Fall für die Entwicklung von Algorithmen und deren Evaluation und Optimierung eingesetzt werden.

Außerhalb des Evaluations-Turnus wurden im Rahmen der Arbeit in den Teilbereichen *Objektdetektion*, *Personenklassifikation*, *Ortsdetektion* sowie *Aktivitätserkennung* domänenspezifische Experimente durchgeführt. Innerhalb der Domäne der *Objektdetektion* ließen sich für die *Personenerkennung* aus dem BBC *EastEnders*-Datensatz 1.000 Schlüsselbilder extrahieren und intellektuell annotieren. Der dabei entstehende Datensatz besteht aus 250 Bildern mit einer Person, 250 Bildern mit zwei Personen und 500 Bildern ohne Personen. In vier Verarbeitungsabläufen konnte mit den Softwareframeworks *Detectron* und *Yolo9000* untersucht werden, ob die Personenerkennung innerhalb von EMSML unabhängig von der verwendeten Technologie durchführbar ist und durch die Verwendung von Verteilungsstrategien in diesem Bereich eine Verbesserung der Performanz erzielt werden kann. Dabei wurden im Rahmen der Verteilung rechnerspezifische und prozessspezifische Performanzwerte sowie die Detektionsgüte erfasst. Diesbezüglich ist erkennbar, dass *Detectron* im Kontext der *Personendetektion* am bereitgestellten Datensatz bessere Ergebnisse als *Yolo9000* erzielen kann und eine höhere *Accuracy* aufweist. Darüber hinaus lassen sich die beiden heterogenen Technologien mittels EMSML einsetzen und durch die Kombination der Ergebnisse Verbesserungen erzielen. Die Fusion auf *Ergebnis-Level* ermöglicht diesbezüglich eine Steigerung der Accuracy von 81 % bei *Detectron* und 72 % für *Yolo9000* auf kumulativ 94 %. Für die Verarbeitung aller 1.00 Bilder wurden mit *Detectron* ca. 100 Sekunden und mit *Yolo9000* ca. 200 Sekunden benötigt. Bei der Verteilung der Verarbeitung auf die beiden Knoten verhielten sich die Knoteneigenen Parameter ähnlich zu den lokalen Versuchen; allerdings wurde die Zeit durch die Parallelverarbeitung signifikant verringert, was auf eine hervorragende Skalierungsfähigkeit von EMSML schließen lässt.

Im Rahmen der *Personenklassifikation* konnte mittels *Decision-Level Fusion* eine Verbesserung der Accuracy von 88 % bei *FaceNet* sowie 88 % bei *OpenFace* auf 95 % bei einer Verarbeitungszeit für 1.000 Bilder von 30 Sekunden für *FaceNet* und 49 Sekunden bei *OpenFace* erreicht werden. Die *Ortsdetektion* basierte auf einem ähnlichen Ansatz, wobei 1.000 Schlüsselbilder von Orten genutzt wurden. Dabei ließen sich mittels Ergebnisfusion der Frameworks *Places365* und *TuriCreate* eine Steigerung der Accuracy von 77 % und 72 % auf 83 % erzielen, wobei die Verarbeitungszeiten 115 respektive 200 Sekunden betrugen.

Experimente im Umfeld der *Aktivitätserkennung* forcierten die Verwendung von Ähnlichkeitsmodellen und einem Bildklassifizierer die auf Einzelbilder von einzelnen Aktivitäten trainiert wurden und sich ebenfalls auf einen Testdatensatz von 1.000 intellektuell zusammengestellten Images anwenden ließen. Dabei konnte eine Accuracy von 69 % und 75 % erreicht werden, wobei Fusionierungsansätze eine Verbesserung dieser auf 77 % ermöglichten. Dabei konnten Verarbeitungszeiten von 215 Sekunden (Ähnlichkeitsmodell) und 60 Sekunden (Bildklassifizierer) erreicht werden.

Die domänenübergreifenden Experimente fokussierten auf die Bearbeitung des Gesamtproblems, wobei im Jahr 2018 Personen an einem Ort und im Folgejahr Personen bei der Ausführung spezifischer Aktivitäten zu identifizieren waren. Dabei sollten jeweils 30 Topics im BBC *EastEnders*-Datensatz gesucht und die Ergebnisse gesammelt werden. Die Übermittlung an das NIST fand anschließend in Form einer XML statt. Diesbezüglich

ermöglichte das EMSML in mehreren Durchläufen im Jahr 2018 3.478 und 3.550 der 11.717 und im Jahr 2019 zwischen 874 bis 939 von 6.592 aller relevanter Schlüsselbilder zu identifizieren. Dabei konnte 2018 eine MAP von maximal 11 % und 2019 eine MAP von maximal 0,82 % in automatisierten Verarbeitungen erzielt werden. Der Einsatz intellektueller Nachbearbeitung ermöglichte 2018 darüber hinaus eine Steigerung der MAP auf 25,2 %. Gesamtheitlich betrachtet ließen sich über beide Jahre 39.186.091 Einzelbilder der 464 Stunden Videomaterialien verarbeiten und über 250 Millionen Metadaten sammeln. In diesem Kontext umfasste die Verarbeitungszeit für die komplexe Problemstellung durchschnittlich zwischen 18 und 24 Tagen sowie die Suchzeit über alle Topics 0,14 *ms* pro Bild.

Ähnlich detailliert wird die Entwicklung in der noch jungen Domäne der Aktivitätsklassifizierung aus multiplen Kameraaufnahmen (ACTEv) innerhalb der TRECVid Evaluationskampagne aufgearbeitet. Erwähnenswert ist, dass die Definition der Ergebnisklassen und der Evaluationsmetrik im Verlauf der ersten beiden Wettbewerbsjahre innerhalb der Evaluationsperioden mehrfach auf die Resultate ausgewählter Teams angepasst wurden, was einen nachhaltigen Systemaufbau bzw. die Erzielung valider vergleichbarer Ergebnisse deutlich erschwerte, weshalb die Arbeit naturgemäß auf die offiziellen Ergebnisse Bezug nimmt und eigene Korrekturen explizit ausweist.

Die Zielstellung der domänenbasierten Experimente war bei ActEV ähnlich ausgerichtet, wie bei INS. Dabei ließen sich die Gebiete *Personen- und Fahrzeugerkennung*, *Tracking* sowie *Aktivitätserkennung* individuell untersuchen.

Im Bereich der *Personen- und Fahrzeugerkennung* konnten aus dem Datensatz VIRAT-V1 1.000 Schlüsselbilder extrahiert und annotiert werden. Die Frameworks *Detectron* und *Yolo9000* beinhalten bereits vortrainierte Modelle, die eine Erkennung von Personen und Fahrzeugen gestattet. Im Rahmen der Experimente stellte sich heraus, dass für den bereitgestellten Datensatz *Detectron* in der Personenerkennung bessere Ergebnisse erzielen konnte, wobei *Yolo9000* hingegen bei der Fahrzeugerkennung bessere Resultate erzielte. Darüber hinaus ermöglichte die *Decision-Level Fusion* eine Genauigkeit von jeweils 90 % auf 93 % zu steigern und die Defizite der beiden Frameworks auszugleichen. Die Verarbeitungszeit aller 1.000 Bilder wurde mit ca. 300 Sekunden für *Detectron* und 500 Sekunden für *Yolo9000* bestimmt. Die im Vergleich zu *Instance Search* längere Bearbeitungszeit lässt sich durch die wesentlich höhere Auflösung von 1920×1080 Pixel des VIRAT-Datensatzes begründen.

Die Experimente im Bereich *Tracking* basierten auf zehn 100 Sekunden langen Videoclips aus dem VIRAT-Datensatz. Die enthaltenen Objekte wurden intellektuell annotiert und die Begrenzungsrahmen aus dieser Annotation mit den (x,y)-Koordinaten der oberen linken Ecke sowie Breite und Höhe gespeichert. Die resultierenden Videoclips wurden anschließend in Einzelbilder zerlegt, wobei sich bewegende Objekte über mehrere Einzelbilder repräsentiert wurden. Dabei konnte ermittelt werden, dass die Gesamtergebnisse im Sinne der Precision 65 %, Recall 75 % und Accuracy 83 % teilweise mittelmäßige bis gute Werte aufweisen, was darauf schließen lässt, dass sich der eigens entwickelte Tracking-Algorithmus für die gestellte Aufgabenstellung eignet. Die intellektuelle Sichtung der Ergebnisse zeigte, dass der Tracker Schwächen bei Überdeckungen und sich verändernder

Geschwindigkeiten von Objekten aufweist. Die Verarbeitung aller 30.000 Bilder dauerte im Durchschnitt 800 Sekunden.

Für die Domäne der *Aktivitätserkennung* wurden aus dem VIRAT-Datensatz weitere 200 Videos mit jeweils 10 Sekunden Länge extrahiert. Dabei konnten die drei Verfahren zur (1) *Posenerkennung via LSTM*, (2) *Extraktion von Bewegungsvektoren* sowie (3 zur) *Größenveränderung der Bounding Boxes* getestet werden, um relevante Aktivitäten automatisiert zu klassifizieren. Der Ansatz der Bewegungsvektoren zeigte in diesem Kontext die besten Ergebnisse im Rahmen von Aktivitäten die auf Bewegungsänderungen fokussierten (bspw. Linksabbiegen, Rechtsabbiegen, Kehrtwende). Der Ansatz der Beobachtung von Größenänderungen von Bounding Boxes gestattet grundsätzlich die Erkennung zur Öffnung eines Kofferraums, wobei dieser Ansatz beim Nachziehen eines Objekts schlechtere Ergebnisse aufwies. Durch Verwendung eines mittels synthetischer Daten trainierten LSTMs ließ sich darüber hinaus eine Aktivitätserkennung von Personen-bezogenen Aktivitäten realisieren. Über alle Aktivitätsklassen hinweg war es damit möglich, eine durchschnittliche Güte von 53 % Precision, 73 % Recall und 71 % Accuracy zu erreichen, wobei die Defizite der einzelnen Ansätze kompensierbar waren. Die Verarbeitungszeiten variieren dabei stark und sind abhängig vom gewählten Ansatz. Der LSTM-Ansatz benötigte erwartungsgemäß doppelt so viel Zeit wie die Ansätze (2) und (3), die mit durchschnittlich $0,023$ Sekunden in etwa gleich lang für die Verarbeitung eines Bildes benötigten.

Basierend auf den Erkenntnissen der domänenspezifischen Experimente ließ sich innerhalb der domänenübergreifenden Experimente das Gesamtproblem bearbeiten. In diesem Kontext waren 2018 das Auftreten von 12 Aktivitäten inklusive Objektpositionen und 2019 ausschließlich 18 Aktivitäten innerhalb der VIRAT-Datensätze zu identifizieren, wobei sich ermittelte Ergebnisse in Form einer JSON an das NIST übermitteln ließen. Dabei war es möglich 3.151.211, bzw. 3.550.215 Personen sowie zwischen 27.030.834 und 27.525.075 Fahrzeuge, innerhalb zugrundeliegender 2.094.413 Einzelbilder der VIRAT-Datensätze in einer Auflösung von 1920x1080px, zu ermitteln. Darüber hinaus konnte gezeigt werden, dass die Objektdetektion bei einer Objekt-P_{miss}-Rate von $0,32\,\%$ gut funktioniert und damit Ergebnisse im Rahmen der Aktivitätserkennung 2018 mit einer P_{miss}-Rate von durchschnittlich $0,94\,\%$ erzielbar waren. Überdies ermöglichte die Optimierung der Ansätze innerhalb des EMSML 2019 das Finden der bisher unerkannten Aktivitäten *Closing* und *Opening*. Weiterhin sank die P_{miss}-Rate der 12 Klassen von 2018 auf $0,91\,\%$, wobei 2019 18 Aktivitäten mit einer P_{miss}-Rate von $0,92\,\%$ ermittelt werden konnten. Über den kompletten Bearbeitungszeitraum beider Jahre ließen sich 90 Millionen Metadaten sammeln. Diesbezüglich umfasste die Verarbeitungszeit zwischen drei und neun Tagen, während die Suchzeit über alle Topics mit durchschnittlich $0,18$ *ms* pro Bild erzielt werden konnte.

Die Arbeit zeigt, dass die Adaption verschiedener moderner Mustererkennungsframeworks als *Single Model Architektur* durch EMSML prinzipiell ermöglicht wird und die Systemleistung durch Einsatz multipler SMAs sowie geeigneter Fusionsmethoden sowohl qualitativ als auch bzgl. der Geschwindigkeit verbessert werden kann. Für den Einblick in weitere relevante Ergebnisse sei auf die einzelnen Kapitel der Arbeit verwiesen.

4 Kapitelübersicht

Die Arbeit gliedert sich in sechs nachfolgend beschriebene Kapitel.

Kapitel 2 umfasst Methoden und Strategien der Bereiche *Maschines Lernen in der Mediendomäne*, *Verteilte Systeme* sowie *Datenmanagement & Virtualisierung*, wobei Definitionen und Begriffserklärungen im Kontext der Arbeit nähere Erläuterung finden. Darauf aufbauend werden aktuelle Technologien und Frameworks zur Detektion von Mustern analysiert, wobei eine Leistungsbewertung erfolgt und ein Kriterienkatalog abgeleitet wird.

In Kapitel 3 ermöglichen Vorüberlegungen die Analyse des zuvor erstellten Kriterienkatalogs hinsichtlich sich daraus ergebender Anforderungen an eine holistische Infrastruktur zur generischen Anwendung von Bilderkennungsalgorithmen im Kontext multipler Problemstellungen in der Domäne der Videoanalyse. Überdies lassen sich Konzepte ableiten, die die Erstellung einer Gesamtarchitektur gestatten. Das so entstehende holistische Framework EMSML bildet die Grundlage für die Umsetzung, Evaluation und Optimierung heterogener Technologien und Ansätze, wobei diesbezüglich ein domänenübergreifender Workflow zur Integration von Single Model Architekturen in die Infrastruktur beschrieben wird. Die so entstehenden Implementationen sind erfahrungsgemäß nur im Zusammenhang mit dem kontextspezifischen Anwendungsfällen konkret untersuchbar, wobei sich wissenschaftliche Evaluationskampagnen, wie beispielsweise TRECVid, aufgrund gegebener Evaluationsmetriken und Datensätze besonders zu eignen scheinen.

Kapitel 4 beinhaltet den ersten Anwendungsfall, das TRECVid-Aufgabengebiets *Instance Search*. In diesem Kontext sind Personen an bestimmten Orten oder bei definierten Aktivitäten zu identifizieren. Dabei wird in einer fachlichen Einordnung die historische Entwicklung des Aufgabengebiets und dessen Abgrenzung zur klassischen Objektdetektion dargestellt und die Spezifika der verwendeten Datensätze und Abgfragetypen analysiert. Weiterhin wird die komplexe Aufgabenstellung in Teilaspekte aufgegliedert und konzeptionelle Lösungsansätze vorgestellt. Anschließend lassen sich die Teillösungen zu einer übergreifenden Lösung kombinieren und aufzeigen, wie sich die gewählten Ansätze mittels der in Kapitel entwickelten Infrastruktur umsetzen lassen. Dabei steht insbesondere die Identifikation und anschließender Anwendung von Optimierungsparametern im Vordergrund, wobei eine Evaluation im Sinne der Funktionalität, Güte und Performanz erfolgt.

Das letzte Anwendungsszenario wird in Kapitel 5 beschrieben und umfasst das TRECVid-Aufgabengebiet *Activities in Extended Video*, wobei Personen oder Fahrzeuge bei der Ausführung definierter Aktivitäten wiederzufinden sind. Durch eine fachliche Einordnung und die Betrachtung der historischen Entwicklung kann dabei eine Abgrenzung zu *Instance Search* aufgezeigt werden. Analog zu Kapitel 4 werden in diesem Kontext zusätzlich die *Related Work* untersucht und der Forschungsfokus sowie die Zielsetzung der eigenen Arbeiten abgeleitet. Darüber hinaus wird aufgezeigt, welche Ansätze sich zur Lösung der komplexen Problemstellung eignen und welche Anpassungen bestehender Technologien zur Bearbeitung des Aufgabenkomplexes notwendig erscheinen. Diesbezüglich lässt sich ebenso eine Evaluation im Sinne der Funktionalität, Güte und Performanz durchführen.

Neben Kapitel 1, das den Anwendungskontext der Arbeit skizziert, ordnet das letzte Kapitel 6 die erzielten Ergebnisse ein und gibt einen Ausblick über zukünftige Verbesserungsmöglichkeiten.

Literaturverzeichnis

[Kü12] Kürsten, J.: A Generic Approach to Component-Level Evaluation in Information Retrieval. Dissertation, Technische Universität Chemnitz, Chemnitz, 2012. ISBN 978-3-941003-68-2.

[Ri14] Ritter, M.: Optimierung von Algorithmen zur Videoanalyse – Ein Analyseframework für die Anforderungen lokaler Fernsehsender. Dissertation, Technische Universität Chemnitz, Chemnitz, 2014. ISBN 978-3-944640-09-9.

[Ro21] Roschke, C.: Generische Verkettung maschineller Ansätze der Bilderkennung durch Wissenstransfer in verteilten Systemen – Am Beispiel der Aufgabengebiete INS und ACTEv der Evaluationskampagne TRECVid. Dissertation, Technische Universität Chemnitz, Chemnitz, 2021. ISBN 978-3-96100-142-2.

Christian Roschke wurde am 19. Mai 1987 in Altdöbern geboren und besuchte von 1999 bis 2006 das Dr.-Albert-Schweitzer Gymnasium in 03226 Vetschau, wo er die Allgemeine Hochschulreife mit der Note 1,8 erlangte. Nach der Schule absolvierte Herr Roschke ein Pflegepraktikum im Evangelischen Krankenhaus Luckau und begann im Sommersemester 2007 mit dem Studium der Medizin an der Justus-Liebig-Universität Gießen. Nach drei Semestern wechselte er die Studienrichtung und nahm ein Bachelor-Studium der *Angewandten Informatik* an der Technischen Universität Chemnitz im Wintersemester 2008 mit dem Schwerpunkt *Eingebettete Systeme* auf, welches er 2012 mit einer 1,7 abschloss. Nach der Erlangung des akademischen Grads *Bachelor of Science* begann Herr Roschke das Master-Studium *Data and Web Engineering* ebenfalls an der Technischen Universität Chemnitz sowie eine Tätigkeit als *Lehrkraft für besondere Aufgaben* (LfBA) an der Hochschule Mittweida. Den *Master of Science* erlangte Herr Roschke im Jahr 2016 mit der Note 1,6. Beruflich führte er seine Tätigkeit als LfBA bis Januar 2017 an der Hochschule Mittweida fort und trat anschließend eine Anstellung als *Akademischer Assistent* im BMBF-Projekt *Stärkung und Erweiterung des akademischen Mittelbaus* (SEM) bis 2020 eben dort an. Während seiner Beschäftigung als Akademischer Assistent begann Herr Roschke 2017 mit der Promotion *Generische Verkettung maschineller Ansätze der Bilderkennung durch Wissenstransfer in verteilten Systemen – Am Beispiel der Aufgabengebiete INS und ACTEv der Evaluationskampagne TRECVid*, die er im September 2021 mit dem Gesamtprädikat *magna cum laudae* verteidigte. 2021 wurde Herr Roschke als Professor für Digitale Transformation & Angewandte Medieninformatik an der Fakultät für Angewandte Computer- und Biowissenschaften der Hochschule Mittweida berufen.

Steffen Hölldobler et al. (Hrsg.): Ausgezeichnete Informatikdissertationen 2021,
Lecture Notes in Informatics (LNI), Gesellschaft für Informatik, Bonn 2022 241

Entwicklungsgeschichte von Genfamilien – Theorie und Algorithmen[1]

David Schaller[2]

Abstract: Das Verständnis der Beziehungen zwischen Genetik und evolutionären Innovationen erfordert die Rekonstruktion der Verwandtschaftsverhältnisse innerhalb von Genfamilien. Dafür werden oft graphentheoretische Methoden eingesetzt. Die Knoten und Kanten der Graphen repräsentieren dabei verwandte Gene bzw. messbare Daten wie die (Un-)Ähnlichkeit der Gensequenzen. Best-Match-Graphen enthalten gerichtete Kanten von jedem Gen zu dessen nächsten Verwandten und sind zentral in der Erkennung von Orthologen (≈ funktional äquivalente Gene der verschiedenen Spezies). Later-Divergence-Time-Graphen werden als erstes formales Modell s.g. impliziter Methoden der Inferenz horizontalen Gentransfers (HGT) präsentiert. Beide Graphenklassen werden charakterisiert. Ihr bisher in der Praxis ungenutztes Potenzial zur Rekonstruktion evolutionärer Szenarien wird algorithmisch zugänglich gemacht und durch Simulationen belegt. Damit liefert diese Arbeit die theoretischen Grundlagen für eine verbesserte automatisierte Erkennung von Orthologen und HGT.

1 Einleitung

Das allseits bekannte Modell zur Beschreibung der Abstammungsgeschichte von Spezies sind phylogenetische Bäume. Demnach stammt eine betrachtete Gruppe von Spezies (die Blätter des Baumes) über eine Reihe von so genannten *Speziationen* (den Verzweigungen) von einem gemeinsamen Vorfahren (der Wurzel) ab. In der Bioinformatik werden Organismen meist durch ihre Genome (die Gesamtheit des Erbmaterials) repräsentiert und infolgedessen durch ihre Gene als zentrale vererbbare Einheiten innerhalb der Genome. Die Evolution der Gene ist eng gekoppelt an die ihrer zugehörigen Spezies. Insbesondere erhalten die resultierenden Spezies einer Speziation je eine Variante des Gens, die danach unabhängig voneinander evolvieren. Allerdings wird die Entwicklungsgeschichte von Genen durch weitere Arten von Ereignissen geprägt. Durch Prozesse wie ungleiches Crossing-over in der Keimzellenbildung können Abschnitte des Genoms samt der darauf befindlichen Gene kopiert werden. Dies kann zu *Duplikationen* führen, d.h. zum Auftreten mehrerer Kopien desselben Gens innerhalb einer Spezies. Auf ähnliche Weise können Gene verloren gehen (*Verluste*). Ein Prozess, der besonders häufig im Reich der Bakterien auftritt, ist zudem *horizontaler Gentransfer (HGT)*. Dabei wird genetisches Material von einer Spezies in eine völlig andere übertragen. Das Ergebnis aller dieser Ereignisse ist ein *Genbaum*, der im Allgemeinen nicht mit dem zugehörigen *Speziesbaum* übereinstimmt (siehe Abb. 1(A) für ein beispielhaftes Szenario mit vier Spezies und acht Genen).

[1] Englischer Titel der Dissertation: "Gene Family Histories – Theory and Algorithms"

[2] Universität Leipzig, Abteilung Bioinformatik, Institut für Informatik, Härtelstr. 16-18, 04107 Leipzig, Deutschland, sdavid@bioinf.uni-leipzig.de

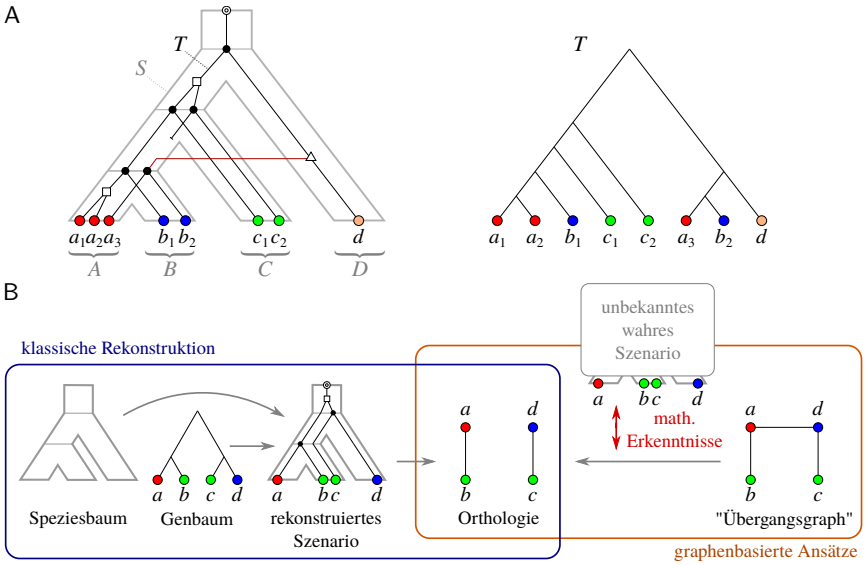


Abb. 1: (A) Ein Evolutionsszenario bestehend aus einem Genbaum T eingebettet in einen Speziesbaum S. Das Szenario enthält Speziationen (●), Duplikationen (□), einen Verlust (⊥) und einen HGT (△). Die Topologie von T (rechts gesondert dargestellt) stimmt nicht mit der von S überein. Insbesondere liegen mehrere verwandte Gene innerhalb derselben Spezies vor (dargestellt durch die gleiche Farbe). (B) Klassische Rekonstruktion von Homologie-Typen (hier Orthologie) vs. graphenbasierte Ansätze.

Entsprechend der verschiedenen Ereignisse in der Evolution von Genen unterscheidet man zwischen Typen von *Homologie* (Verwandtschaft von Genen): Zwei Gene sind *Orthologe* bzw. *Paraloge*, wenn sie sich infolge einer Speziation bzw. Duplikation auseinanderentwickelt haben. In Abb. 1(A) sind zum Beispiel a_1 und b_2 Orthologe, da ihr letzter gemeinsamer Vorfahre (*LCA*, von engl. *last common ancestor*) mit einer Speziation gekennzeichnet ist. Dagegen sind a_1 und a_2 Paraloge, aber auch die zu verschiedenen Spezies gehörenden Gene a_1 und c_2. Schließlich bezeichnet man zwei Gene als *Xenologe* wenn ihre Geschichte seit dem LCA einen HGT enthält, wie es u.a. bei a_3 und d der Fall ist. Die Kenntnis dieser Homologie-Relationen ist von zentraler Bedeutung innerhalb der biologischen Forschung. Orthologe Gene weisen in der Regel eine höhere Übereinstimmung in ihren Funktionen auf als Paraloge [TKL97] und werden daher genutzt um Genfunktionen in Modellorganismen zu erforschen und neu sequenzierte Genome zu annotieren. Xenologie und horizontaler Gentransfer spielen u.a. eine wichtige Rolle bei der Entstehung von Antibiotikaresistenzen vor Erregern sowie in Mikroorganismen-Gemeinschaften zahlreicher ökologischer Systeme [SHG15]. Im Allgemeinen sind die Details evolutionärer Szenarien die Voraussetzung um evolutionäre Prozesse insbesondere auf genetischer Ebene zu verstehen. Die Relevanz dieser Forschung hat sich zuletzt eindrucksvoll im Kampf gegen die Varianten von Covid-19 gezeigt.

Aus diesen Gründen wurde ein Spektrum an Methoden entwickelt um Homologie-Relationen aus verfügbaren biologischen Daten, im einfachsten Fall den Sequenzen der verwandten Gene, zu inferieren [AGD19; Ra15]. Klassische Ansätze verwenden hierzu in der Regel probabilistische Methoden wie etwa *Maximum Likelihood* um sowohl einen Genbaum T als auch einen Speziesbaum S explizit zu konstruieren [Fe04]. In einem zweiten Schritt wird dann anhand bestimmter Optimalitätskriterien eine sogenannte *Übereinstimmungsabbildung* konstruiert, d.h. eine Einbettung von T in S, aus welcher dann direkt die Homologie-Relationen abgeleitet werden können [AGD19], siehe Abb. 1(B). Solche Ansätze sind in der Regel sehr rechenintensiv und daher auf kleine Datensätze beschränkt. Fortschritte beim Verständnis der mathematischen Eigenschaften von Evolutionsszenarien und den Homologie-Relationen haben die Entwicklung neuer, i.d.R. schnellerer Methoden angetrieben, die das Szenario nicht explizit rekonstruieren und außerdem deutlich weniger Modellannahmen erfordern [AGD19; He15]. Stattdessen werden aus den biologischen Daten Graphen auf der Menge von Genen konstruiert, in denen eine Kante (a, b) beispielsweise kodiert, dass b das ähnlichste Gen in seiner Spezies zu Gen a ist. Theoretische Erkenntnisse aus dem Zusammenhang zwischen dem unbekannten wahren Szenario und den dadurch erklärbaren Homologie-Relationen werden dann angewandt um diesen "Übergangsgraphen" beispielsweise in einen Graphen zu überführen, in dem Kanten Orthologie repräsentieren, siehe ebenfalls Abb. 1(B).

In dieser Arbeit [Sc21] werden die mathematischen Grundlagen für zwei solcher *graphen-basierten* Ansätze erarbeitet: Best-Match-Graphen (Abschnitt 3) und Later-Divergence-Time-Graphen (Abschnitt 4). Insbesondere wird ihr Potenzial aber auch ihre Limitationen im Hinblick auf die Rekonstruktion von Evolutionsszenarien theoretisch untersucht und mithilfe neu entwickelter Polynomialzeit-Algorithmen und Simulationen empirisch belegt.

2 Mathematische Definitionen und Notation

Ungerichtete bzw. *gerichtete Graphen* $G = (V, E)$ bestehen aus einer Knotenmenge $V(G) := V$ und einer Menge von ungerichteten Kanten $xy = yx \in E(G) := E \subseteq \binom{V}{2}$ bzw. gerichteten Kanten $(x, y) \in E(G) := E \subseteq (V \times V) \setminus \{(v, v) \mid v \in V\}$. Eine *Färbung* eines Graphen ist eine Abbildung $\sigma\colon V \to M$ mit einer Farbenmenge M und heißt *zulässig* falls $xy \in E$ (bzw. $(x, y) \in E$) $\implies \sigma(x) \neq \sigma(y)$ für alle $x, y \in V$.

> *Hier repräsentiert ein Knoten x i.d.R. ein Gen und die Farbe $\sigma(x)$ die Spezies, in deren Genom x enthalten ist.*

(Gewurzelte) *Bäume* sind zusammenhängende, kreisfreie, ungerichtete Graphen $T = (V, E)$ mit einem ausgezeichneten Knoten ρ_T, der *Wurzel* genannt wird. Die Wurzel induziert eine Halbordnung $\preceq_T$ (die *Vorfahren-Ordnung*) auf V entsprechend $x \preceq_T y$ g.d.w. y auf dem Pfad von x zur Wurzel ρ_T liegt. Die $\preceq_T$-minimalen Elemente in V heißen *Blätter* und

werden mit L bzw. $L(T)$ bezeichnet. Es werden häufig Bäume (T, σ) mit Blätterfärbung $\sigma\colon L \to M$ betrachtet. Für zwei Knoten $x, y \in V$ ist der *letzte gemeinsame Vorfahre*, $\mathrm{LCA}_T(x, y)$, der eindeutige $\preceq_T$-minimale Knoten $v \in V$ sodass $x \preceq_T v$ und $y \preceq_T v$ gilt. Ein Tripel $t = ab|c$ ist ein Baum mit drei Blättern a, b und c und zwei weiteren Knoten sodass $\mathrm{LCA}_t(a, b) \prec_t \mathrm{LCA}_t(a, c)$. Ein Baum T *zeigt* das Tripel $ab|c$ falls $a, b, c \in L(T)$ und $\mathrm{LCA}_T(a, b) \prec_T \mathrm{LCA}_T(a, c)$. Eine Menge $\mathcal{R}$ von Tripeln heißt *konsistent* falls ein Baum existiert, der alle Tripel in $\mathcal{R}$ zeigt. Konsistenz einer Tripelmenge $\mathcal{R}$ (definiert auf einer Blattmenge L) kann mit dem Algorithmus `BUILD` in Polynomialzeit überprüft werden [Ah81]. Im positiven Fall konstruiert `BUILD` einen eindeutigen Baum $\mathrm{Aho}(\mathcal{R}, L)$ der alle Tripel in $\mathcal{R}$ zeigt.

3 Best-Match-Graphen und Orthologie

3.1 Grundlagen und Charakterisierung

Der erste Schritt in der graphenbasierten Inferenz von Orthologie besteht in der Regel darin, für jedes Gen x und für jede Spezies $Y \neq \sigma(x)$ die *Best Matches* zu bestimmen, d.h. die nächsten Verwandten von x in Spezies Y. Diese werden in der Regel dadurch approximiert, indem die Gene mit der höchsten Sequenzähnlichkeit etwa mithilfe von Software wie `blast` identifiziert werden [AGD19]. Obwohl Vertreter dieser Methoden schon länger erfolgreich eingesetzt werden, wurden Best Matches und Best-Match-Graphen erst kürzlich von Geiß et al. [Ge19] formal definiert:

Definition 1 *Sei (T, σ) ein blättergefärbter Baum. Dann ist $y \in L$ ein* Best Match *von $x \in L$, in Symbolen $x \to y$, falls*
(i) $\sigma(x) \neq \sigma(y)$ und
(ii) $\mathrm{LCA}_T(x, y) \preceq_T \mathrm{LCA}_T(x, y')$ für alle $y' \in L$ der Farbe $\sigma(y') = \sigma(y)$ gilt.
Falls außerdem $y \to x$ gilt, dann heißen x und y reziproke Best Matches. *Der gefärbte, gerichtete Graph* $\mathrm{BMG}(T, \sigma)$ *mit Knotenmenge L und Kanten (x, y) g.d.w. y ein Best Match von x ist heißt* Best-Match-Graph (BMG) *von (T, σ).*

Umgekehrt ist ein gefärbter, gerichteter Graph $(\vec{G}, \sigma)$ ein BMG, wenn er von einem Baum (T, σ) *erklärt* wird, d.h. sodass $(\vec{G}, \sigma) = \mathrm{BMG}(T, \sigma)$ gilt. Abb. 2 zeigt beispielhaft einen Baum (T, σ) und den zugehörigen Graphen $\mathrm{BMG}(T, \sigma)$. Da der Genbaum in praktischen Anwendungen nicht *a priori* bekannt ist, erhält man aus den Sequenzvergleichen im Allgemeinen einen gerichteten Graphen, der nicht notwendigerweise ein BMG im Sinne von Definition 1 ist. Vom theoretischen Standpunkt her stellt sich daher zunächst das Problem der Erkennung von BMGs, d.h. nach der Frage ob ein erklärender Baum (T, σ) zu einem gegebenen Graphen $(\vec{G}, \sigma)$ existiert. Um dies zu beantworten ist die Beobachtung hilfreich, dass BMGs Informationen über die Topologie aller möglichen Bäume enthalten, die sie erklären. Angenommen $(\vec{G}, \sigma)$ wird von einem Baum (T, σ) erklärt und $\vec{G}$ enthält die Kante

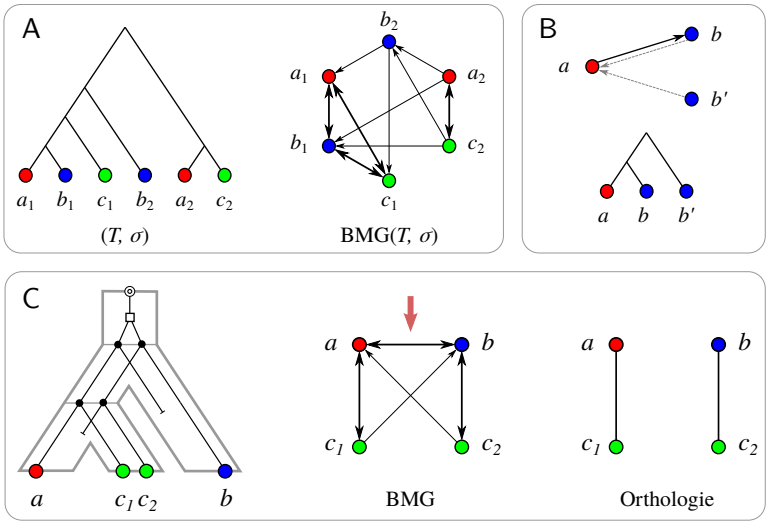


Abb. 2: (A) Ein blättergefärbter Baum (T, σ) mit zugehörigem Best-Match-Graphen BMG(T, σ). Die reziproken Best Matches sind als dickere Pfeile hervorgehoben. (B) Der oben darstellte induzierte Teilgraph (gestrichelte Kanten sind optional) induziert das darunter dargestellte informative Tripel $ab|b'$. (C) Ein Duplikation-Verlust-Szenario mit zugehörigem BMG. Die markierten reziproken Best Matches a und b sind sichere Falsch-Positive (*sFP*) bzgl. Orthologie (rechts dargestellt).

(a, b) jedoch nicht die Kante (a, b') wobei b und b' von der gleichen Spezies stammen und a von einer anderen (siehe Abb. 2(B) oben). Dann folgt aus der Definition der Best Matches, dass b näher verwandt mit a sein muss als b' und damit dass T das Tripel $ab|b'$ zeigen muss. Somit kann aus $(\vec{G}, \sigma)$ eine Menge

$$\mathcal{R}(\vec{G}, \sigma) := \{ab|b' \colon \sigma(a) \neq \sigma(b) = \sigma(b'),\ (a, b) \in E(\vec{G}) \text{ und } (a, b') \notin E(\vec{G})\} \quad (1)$$

von *informativen Tripeln* konstruiert werden, die von *jedem* Baum gezeigt werden, der $(\vec{G}, \sigma)$ erklärt. Im Fall, dass ein solcher Baum existiert, muss $\mathcal{R}(\vec{G}, \sigma)$ konsistent sein. Diese Bedingung ist allein allerdings nicht hinreichend. Der `BUILD`-Algorithmus ermöglicht die Konstruktion eines Kandidatenbaumes $(\hat{T}, \sigma)$, dessen BMG mit dem gegebenen Graphen verglichen werden kann. Die Gleichheit BMG$(\hat{T}, \sigma) = (\vec{G}, \sigma)$ ist notwendig und *per definitionem* hinreichend dafür, dass $(\vec{G}, \sigma)$ ein BMG ist [Sc21, **Theorem 4.2**]. Insbesondere ist $(\hat{T}, \sigma)$ der eindeutige *minimal aufgelöste* Baum von $(\vec{G}, \sigma)$, d.h. es können keine Kanten in $(\hat{T}, \sigma)$ kontrahiert werden können, ohne dass sich der erklärte BMG ändert. Seine Eindeutigkeit impliziert, dass dieser Baum immer eine kantenkontrahierte Version des wahren Genbaums ist. Dies ergibt sich allein aus der Kombinatorik der Best Matches ohne zusätzliche Annahmen über die evolutionäre Geschichte, die nicht von den Daten gestützt werden. Damit stellen BMGs eine robuste Alternative zu den klassischen Methoden der Genbaum-Rekonstruktion dar, die auf probabilistischen Modellen zur Evolution von Sequenzen beruhen [Fe04].

Wie bereits angedeutet sind die in der Praxis erhaltenen Graphen oft keine BMGs. Dies ist einerseits die Folge einer variablen Mutationsrate zwischen unterschiedlichen Zweigen der Genfamilie, welche wiederum dazu führen kann, dass nächste Verwandtschaft und höchste Sequenzähnlichkeit nicht in allen Fällen übereinstimmen. Solche systematischen Einflüsse können durch Einbeziehung sogenannter "Außengruppen-Gene" korrigiert werden [St20]. Auf der anderen Seite sind zufällige Fehler und Rauschen in Analysen auf biologischen Daten in der Regel unvermeidbar. Ein informatischer Ansatz, die dadurch verbleibenden Fehler in den approximierten Graphen zu korrigieren, besteht darin möglichst wenige Kanten zu entfernen (BMG DELETION), hinzufügen (BMG COMPLETION) oder beides (BMG EDITING) um einen validen BMG zu erhalten. Die Entscheidungsversionen dieser Optimierungsprobleme sind NP-vollständig [Sc21, **Theorem 5.5**]. Viele Probleme in der Bioinformatik sind in Polynomialzeit nicht exakt lösbar (falls $P \neq NP$) und werden in der Praxis durch effiziente Heuristiken gelöst [Fe04]. In [Sc21, Kapitel 5] werden sowohl eine ILP-Formulierung für das exakte Lösen des BMG EDITING-Problems als auch eine Klasse von Heuristiken beschrieben. Empirische Analysen zu Letzteren zeigen, dass BMG EDITING auch für größere Eingabegraphen praktikabel ist.

3.2 Inferenz von Orthologie

Nachdem der aus Gensequenzdaten approximierte Graph zu einem validen BMG $(\vec{G}, \sigma)$ korrigiert wurde, besteht der nächste Schritt darin, die Orthologie-Relation zu extrahieren. Da die Kenntnis von Gen- und Speziesbäumen im Rahmen der hier betrachteten Methoden nicht angenommen wird, müssen alle möglichen Szenarien in Betracht gezogen werden, die $(\vec{G}, \sigma)$ erklären, um sichere Aussagen über das unbekannte wahre Szenario machen zu können. In Duplikation-Verlust-Szenarien (d.h. HGT wird ausgeschlossen) ist dies implizit in Polynomialzeit möglich. Insbesondere sind alle "wahren Orthologen" auch reziproke Best Matches [Ge20]. Es können daher nur Falsch-Positive unter den letzteren auftreten (z.B. a und b in Abb. 2(C)). Reziproke Best Matches a und b in $(\vec{G}, \sigma)$ heißen *sichere Falsche-Positive (*sFP*)*, falls sie in *keinem* Szenario, das $(\vec{G}, \sigma)$ erklärt, Orthologe sind. In [Sc21, Kapitel 6] werden diese sicheren falschen Orthologie-Zuweisungen in BMGs charakterisiert und ein Polynomialzeit-Algorithmus für ihre Identifizierung wird vorgestellt. Die Relevanz für praktische Anwendungen, die bisher keine theoretisch fundierte *sFP*-Korrektur enthalten, wurde mithilfe von Simulationen gezeigt. Die Software `AsymmeTree`[3] generiert zufällige realistische Szenarien, aus den sowohl der BMG als auch die wahre Orthologie-Relation zum Zweck des Benchmarkings extrahiert werden können. Abb. 3 zeigt die drastische Verbesserung der False Discovery Rate durch die Eliminierung der *sFP* aus den reziproken Best Matches. Für die große Mehrheit der simulierten Szenarien kann die Orthologie-Relation auf diese Weise perfekt aus dem BMG rekonstruiert werden. Die in einigen Szenarien verbleibenden Falsch-Positiven können durch die Kenntnis der Best Matches allein nicht identifiziert werden.

[3] `https://github.com/david-schaller/AsymmeTree`

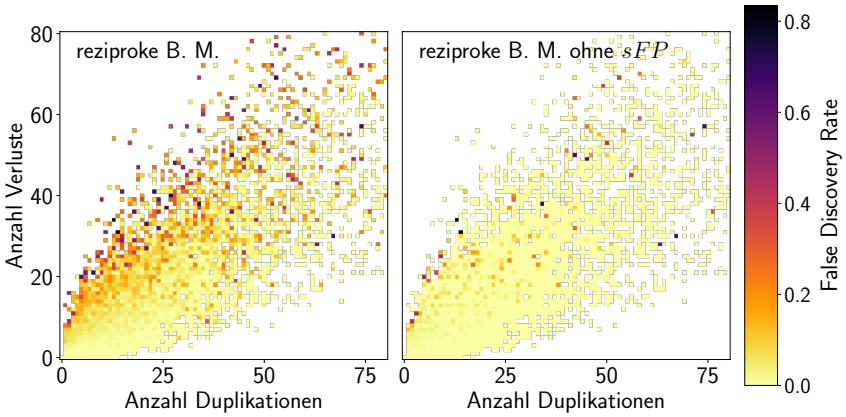


Abb. 3: Verbesserung der Orthologie-Erkennung durch Entfernung aller sicheren Falsch-Positiven (*sFP*). Dargestellt sind Mittelwerte der False Discovery Rate in Abhängigkeit von der Anzahl an Duplikations- und Verlustereignissen für 25.000 simulierte Szenarien. Links wurden die reziproken Best Matches direkt verwendet, rechts wurden alle *sFP* entfernt.

4 Later-Divergence-Time-Graphen und Xenologie

Eine Möglichkeit, die im letzten Abschnitt beschriebenen Methoden auch für Genfamilien anzuwenden, die HGT enthalten, besteht darin zunächst die Xenologen zu bestimmen und dann HGT-freie Teilszenarien zu betrachten. Sogenannte *implizite Methoden der HGT-Inferenz* identifizieren Genpaare, die näher miteinander verwandt sind als es für die zugehörigen Spezies zu erwarten wäre [Ra15]. Die Idee dahinter ist, dass sich eine solche Situation nur durch HGT erklären lässt. Im Rahmen dieser Arbeit wurde das erste formale mathematische Modell für diese Gruppe von Methoden entwickelt, die Later-Divergence-Time-Graphen (LDT-Graphen), welche ähnlich wie BMGs über das (in der Praxis unbekannte) Szenario $\mathcal{S} = (S, T, \sigma, \tau)$ definiert sind. Dieses besteht aus einem Speziesbaum S, einem blättergefärbten Genbaum (T, σ) und einer Datierungsfunktion τ für die Knoten der Bäume (siehe Abb. 4).

Definition 2 *Sei $\mathcal{S} = (S, T, \sigma, \tau)$ ein Evolutionsszenario. Der ungerichtete, gefärbte LDT-Graph $G_<(\mathcal{S})$ hat die Knotenmenge $L(T)$ und eine Kante xy g.d.w. $\tau(\mathrm{LCA}_T(x, y)) < \tau(\mathrm{LCA}_S(\sigma(x), \sigma(y)))$.*

Mit anderen Worten verbinden LDT-Graphen genau diejenigen Gene mit einer Kante, die näher verwandt sind als ihre zugehörigen Spezies. LDT-Graphen können charakterisiert werden als zulässig gefärbte Vertreter einer gut untersuchten Graphenklasse, den Cographen, die zusätzlich eine bestimmte Konsistenzbedingung erfüllen, die sich aus der Färbung ableitet [Sc21, **Theorem 7.3**]. Das Modell LDT-Graph ist konsistent mit der Idee hinter

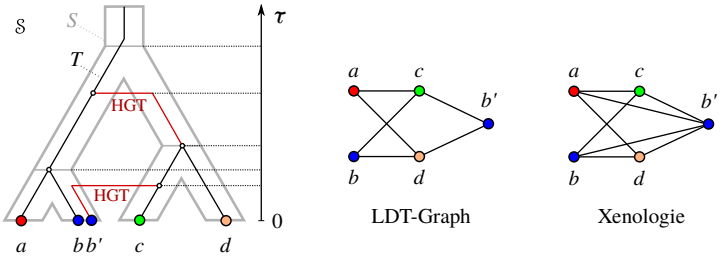


Abb. 4: Ein Szenario $\mathcal{S} = (S, T, \sigma, \tau)$ mit zwei HGT-Ereignissen sowie zugehörigem LDT-Graphen und Xenologie-Relation.

den bereits in der Praxis angewandten Methoden, d.h. Gene die später divergierten als die Spezies in denen sie zu finden sind, sind immer Xenologe [Sc21, **Theorem 8.1**]. Wie das Beispiel in Abb. 4 zeigt, gilt die Umkehrung jedoch im Allgemeinen nicht. Daraus ergibt sich die Aufgabe die fehlenden Xenologenpaare zu inferieren. Ein möglicher Ansatz, der dem in der Bioinformatik häufig angewandten Parsimonitäts-Paradigma folgt, besteht darin, eine minimale Kantenmenge im LDT-Graphen zu ergänzen, sodass dieser einer validen Xenologie-Relation [He18] entspricht. In [Sc21, Kapitel 8] wird ein Polynomialzeit-Algorithmus für die Lösung dieses Optimierungsproblem präsentiert.

Die Relevanz dieses Ansatzes wurde mithilfe von Simulationen demonstriert, die ein breites Spektrum an Parametern abdecken (Raten für Duplikations-, Verlust- und HGT-Ereignisse, etc.). Während die LDT-Graphen im Median nur etwa ein Viertel der tatsächlichen Xenologenpaare als Kanten enthalten, kann der Recall durch die minimale Xenologie-Ergänzung im Median auf 60% bis über 90% gesteigert werden (je nach Wahl der Parameter, vgl. [Sc21, Abb. 78]). Im Gegensatz zum ursprünglichen LDT-Graphen können die Ergebnisse der minimalen Ergänzung Falsch-Positive enthalten, was jedoch nur selten aufzutreten scheint. So wurde für alle betrachteten Parameterkombinationen eine Precision von (im Median) über 95% beobachtet. Zusammenfassend offenbaren diese Ergebnisse das bisher ungenutzte Potenzial der impliziten Methoden zur HGT-Erkennung einen Großteil der verbleibenden Xenologenpaare zu inferieren.

5 Schlussbemerkungen

Diese Arbeit ist ein Beitrag zum Verständnis der Entwicklung von Genfamilien im Zusammenspiel mit der Evolution der Spezies und weiterhin zur Etablierung und Verbesserung neuer bzw. bestehender graphentheoretischer Ansätze für die Inferenz von Homologie-Relationen. Best-Matchen-Graphen und verschiedene Unterklassen wurden charakterisiert und effiziente Algorithmen zur Konstruktion der mit ihnen assoziierten Genbäume entwickelt. Eine exakte ILP-basierte Methode zur Lösung des NP-schweren BMG EDITING-Problems sowie eine Klasse tripelbasierter Heuristiken (mit sieben konkreten Varianten) wurden präsentiert. Letztere machen die Korrektur der aus Sequenzdaten gewonnenen Graphen

zu BMGs praktikabel. Der zweite Schritt einer Inferenz-Pipeline für Orthologie, welche auf dem neu etablierten, formalen Modell der BMGs basiert, ist die Identifizierung der Falschzuweisungen unter den reziproken Best Matches – ein Problem, das in HGT-freien (Teil-)Szenarien ebenfalls effizient und ohne die explizite Konstruktion der möglichen Szenarien gelöst werden kann. In Simulationen wurden auf die Weise die Orthologie-Relationen aus BMGs nahezu perfekt inferiert. Des Weiteren liegt hier das erste formalisierte Modell für implizite Methoden der HGT-Inferenz vor. Es wurde demonstriert wie auf Basis dieses Modells der Anteil der korrekt erkannten Xenologenpaare drastisch erhöht werden kann. Selbstverständlich limitieren Ungenauigkeiten bei der Approximation der BMGs und LDT-Graphen aus Sequenzdaten diese theoretischen Ereignisse in gewissem Maße. Der nächste Schritt besteht daher darin, die Erkenntnisse in eine Software-Pipeline umzusetzen, umfangreichere Tests mit biologischen Daten durchzuführen und die Anwendung schließlich für großangelegte Analysen von Genfamilien durch Biologen nutzbar zu machen. Da sowohl BMGs als auch LDT-Graphen wertvolle Informationen über die zugrundeliegenden Szenarien enthalten, die über rein kombinatorische Methoden zugänglich sind, ist auch eine Komplementierung mit den klassischen (probabilistischen) Methoden der Phylogenetik ein vielversprechender Ansatzpunkt für zukünftige Forschung.

Literatur

[AGD19] Altenhoff, A. M.; Glover, N. M.; Dessimoz, C.: Inferring Orthology and Paralogy. In (Anisimova, M., Hrsg.): Evolutionary Genomics. Bd. 1910, Springer New York, New York, NY, S. 149–175, 2019, ISBN: 978-1-4939-9073-3 978-1-4939-9074-0.

[Ah81] Aho, A. V.; Sagiv, Y.; Szymanski, T. G.; Ullman, J. D.: Inferring a tree from lowest common ancestors with an application to the optimization of relational expressions. SIAM Journal on Computing 10/, S. 405–421, 1981.

[Fe04] Felsenstein, J.: Inferring phylogenies. Sinauer Associates, Sunderland, Mass, 2004, ISBN: 978-0-87893-177-4.

[Ge19] Geiß, M.; Chávez, E.; González Laffitte, M.; López Sánchez, A.; Stadler, B. M. R.; Valdivia, D. I.; Hellmuth, M.; Hernández Rosales, M.; Stadler, P. F.: Best Match Graphs. Journal of Mathematical Biology 78/, S. 2015–2057, 2019.

[Ge20] Geiß, M.; González Laffitte, M. E.; López Sánchez, A.; Valdivia, D. I.; Hellmuth, M.; Hernández Rosales, M.; Stadler, P. F.: Best Match Graphs and Reconciliation of Gene Trees with Species Trees. Journal of Mathematical Biology 80/, S. 1459–1495, 2020.

[He15] Hellmuth, M.; Wieseke, N.; Lechner, M.; Lenhof, H.-P.; Middendorf, M.; Stadler, P. F.: Phylogenomics with paralogs. Proceedings of the National Academy of Sciences 112/, S. 2058–2063, 2015.

[He18] Hellmuth, M.; Long, Y.; Geiß, M.; Stadler, P. F.: A Short Note on Undirected Fitch Graphs. The Art of Discrete and Applied Mathematics 1/, P1.08, 2018, ISSN: 2590-9770.

[Ra15] Ravenhall, M.; Škunca, N.; Lassalle, F.; Dessimoz, C.: Inferring Horizontal Gene Transfer. PLOS Computational Biology 11/, e1004095, 2015, ISSN: 1553-7358.

[Sc21] Schaller, D.: Gene Family Histories - Theory and Algorithms, Dissertation, Universität Leipzig, 2021, URL: https://nbn-resolving.org/urn:nbn:de:bsz:15-qucosa2-763963.

[SHG15] Soucy, S. M.; Huang, J.; Gogarten, J. P.: Horizontal gene transfer: building the web of life. Nature Reviews Genetics 16/, S. 472–482, 2015, ISSN: 1471-0056, 1471-0064.

[St20] Stadler, P. F.; Geiß, M.; Schaller, D.; López Sánchez, A.; Gonzalez Laffitte, M.; Valdivia, D. I.; Hellmuth, M.; Hernández Rosales, M.: From pairs of most similar sequences to phylogenetic best matches. Algorithms for Molecular Biology 15/, S. 5, 2020.

[TKL97] Tatusov, R. L.; Koonin, E. V.; Lipman, D. J.: A Genomic Perspective on Protein Families. Science 278/, S. 631–637, 1997, ISSN: 00368075, 10959203.

David Schaller wurde am 1. Juni 1994 in Glauchau (Deutschland) geboren. Er studierte Biologie (*B. Sc.*, 2017) und Bioinformatik (*M. Sc.*, 2019) an der Universität Leipzig. Im Zeitraum von November 2019 bis Juni 2021 absolvierte er ein Promotionsstudium am Max-Planck-Institut für Mathematik in den Naturwissenschaften. Dort arbeitete er im Bereich *Diskrete Biomathematik* in enger Kooperation mit Prof. Dr. Peter F. Stadler (Uni Leipzig) und Prof. Dr. Marc Hellmuth (Uni Stockholm). Die Universität Leipzig verlieh ihm im Oktober 2021 den Doktorgrad mit dem Prädikat *summa cum laude*. Seit Juli 2021 ist er wissenschaftlicher Mitarbeiter der Bioinformatik-Gruppe der Universität Leipzig. Seine aktuellen Forschungsinteressen liegen im Bereich Phylogenetik, insbesondere in der Entwicklung von mathematischen Methoden und Algorithmen zur Rekonstruktion evolutionärer Szenarien aus verfügbaren biologischen Daten.

Auswirkungen der technologischen Unterstützung auf die Arbeitsbelastung beim Software-Prototyping[1]

Sarah Suleri[2]

Abstract: Prototyping ist ein iteratives Verfahren zur Ideenfindung, Kommunikation und Bewertung von Benutzeroberflächendesigns. Diese Forschung zielt darauf ab, diesen Prozess unter drei Aspekten zu analysieren: traditionelles Prototyping, Rapid Prototyping und Prototyping für Barrierefreiheit. Wir schlagen drei neue Ansätze vor und setzen sie durch die Einführung von drei Artefakten um: 1) Eve, eine skizzenbasierte Prototyping-Workbench, die die automatisierte Umwandlung von Low-Fidelity-Prototypen in höhere Fidelitys unterstützt, 2) Kiwi, eine UI-Design-Pattern- und Richtlinien-Bibliothek zur Unterstützung von UI-Design-Pattern-getriebenem Prototyping, 3) Personify, eine Persona-basierte UI-Design-Richtlinien-Bibliothek für barrierefreies UI-Prototyping. Empirische Untersuchungen mittels NASA-TLX zeigen massive Zeiteinsparungen und deutlich reduzierte subjektive Arbeitsbelastung durch die vorgestellten Unterstützungstools.

1 Einführung

Benutzeroberfläche (UI) Prototyping[3] ist ein iterativer Prozess, der es ermöglicht es UI/UX-Designern, interaktive Modelle ihrer UI-Designs zu erstellen. Diese Prototypen können zum Brainstorming verschiedener Lösungen, zur Kommunikation von Designideen mit Kollegen und zur weiteren Bewertung mit Experten und Endbenutzern verwendet werden [Ca17]. Während des gesamten Softwareentwicklungsprozesses können UI-Prototypen verschiedene Zwecke erfüllen: ein *Analyse-Artefakt*, um den Problemraum mit den Beteiligten zu erkunden, ein *Anforderungs-Artefakt*, um die anfängliche Vision des Systems darzustellen, ein *Design-Artefakt*, um den Lösungsraum des Systems zu erkunden, ein *Kommunikations-Artefakt*, um verschiedene mögliche UI-Designs des Systems zu teilen und zu diskutieren. In vielen Fällen dienen UI-Prototypen als potenzielle Grundlage für die Erstellung des eigentlichen Endprodukts.

UI-Prototypen entwickeln sich in drei Stufen: Low-Fidelity (lo-fi), Medium-Fidelity (me-fi) und High-Fidelity (hi-fi). Der Unterschied zwischen den drei Stufen kann in Bezug auf die Reife des UI-Designs und der Interaktivität betrachtet werden. Ein lo-fi-Prototyp kann eine grobe Freihandskizze oder ein Papierprototyp sein, me-fi - ein digitaler Entwurf, der auf den lo-fi-Skizzen basiert, und hi-fi - ein verfeinerter interaktiver Prototyp, der dem Endprodukt sehr ähnlich ist.

In dieser Dissertation [Su21] wollten wir den Software-Prototyping-Prozess eingehend untersuchen, um den Arbeitsablauf und die Probleme der Designer zu analysieren. Wir un-

[1] English title of the dissertation: "Impact of technological support on the workload of software prototyping"
[2] RWTH Aachen University, sarah.suleri@rwth-aachen.de
[3] In dieser Dissertation werden die Begriffe *Software Prototyping* und *Benutzeroberfläche (UI) Prototyping* synonym verwendet.

tersuchten den Prototyping-Prozess aus der Perspektive der Arbeitsbelastung. Der Begriff Arbeitsbelastung bezieht sich hier auf die subjektive Wahrnehmung des Grades der physischen und kognitiven Belastung, die der Designer während des gesamten Prototyping-Prozesses erfährt. In Anbetracht der Tatsache, dass verschiedene Menschen unterschiedliche Fähigkeiten und Fertigkeiten haben, untersuchten wir auch verschiedene Methoden zur Berechnung der subjektiven Arbeitsbelastung beim Prototyping, die für ein breites Spektrum von Benutzern verallgemeinerbar sind. Unser Ziel war es, den UI-Prototyping-Prozess technologisch zu unterstützen und die Auswirkungen dieser technologischen Unterstützung auf die subjektive Arbeitsbelastung zu vergleichen, die die Designer in den traditionellen (*as-is*) und technologisch unterstützten (*to-be*) Szenarien erfahren.

2 Thesis Statement & Forschungsfragen

Diese Dissertation untersucht die subjektive Arbeitsbelastung während des Prototyping-Prozesses und wie sich die technische Unterstützung dieses Prozesses auf diese Arbeitsbelastung auswirkt.

Im Einzelnen erforschten wir die folgenden Forschungsfragen:

RQ1: Wie hoch ist der Arbeitsaufwand für das Software-Prototyping?
RQ2: Wie können wir das Software-Prototyping technologisch unterstützen?
RQ3: Wie wirkt sich diese technologische Unterstützung für das Software-Prototyping auf die Arbeitsbelastung aus?

Diese Forschung fokussierte beispielhaft das UI-Prototyping für Smartphone-Anwendungen. Die formativen Untersuchungen, die vorgeschlagenen Lösungen und ihre jeweiligen Bewertungen sind entsprechend ausgerichtet.

3 Forschungsansatz

Wir haben uns bei unserer Forschung an den nutzerzentrierten Ansatz gehalten [IS10]. Der Prozess des nutzerzentrierten Designs (User-centered Design, UCD) unterstützt die Forschung und das Design auf der Grundlage eines genauen Verständnisses der Zielnutzer, ihrer Aufgaben und ihrer natürlichen Umgebung. UCD befasst sich mit der gesamten Benutzererfahrung und bezieht die Benutzer während des gesamten Design- und Entwicklungsprozesses mit ein. Es handelt sich um einen iterativen Prozess, der von den Zielnutzern vorangetrieben, verfeinert und bewertet wird.

Dem UCD-Ansatz folgend, haben wir das Software-Prototyping unter drei verschiedenen Aspekten analysiert: traditionelles Prototyping, Rapid Prototyping und Prototyping für Barrierefreiheit. Für jeden Aspekt begannen wir unsere Untersuchung mit halbstrukturierten Interviews mit UI/UX-Designern, um ihre Arbeitsabläufe, aktuellen Praktiken, bevorzugten Tools und Problembereiche zu verstehen.

Nach einer sorgfältigen Analyse schlugen wir Lösungen für jeden Aspekt vor und bewerteten ihre Nutzbarkeit mit UI/UX-Designern. Darüber hinaus untersuchten wir die Auswirkungen der Verwendung dieser neuartigen Lösungen auf die subjektive Arbeitsbelastung von UI/UX-Designern während des Software-Prototyping.

4 RQ1: Wie hoch ist der Arbeitsaufwand für das Software Prototyping?

Um das *as-is*-Szenario zu untersuchen, haben wir die subjektive Arbeitsbelastung von 18 UI/UX-Designern während des Software-Prototypings mit Hilfe des NASA Task Load Index (NASA-TLX) [Go10] untersucht. Die Teilnehmer wurden gebeten, eine zufällig zugewiesene Anwendung zu prototypisieren und ihre subjektive Wahrnehmung der erlebten Arbeitsbelastung für jede Fidelity anzugeben. Die Ergebnisse zeigen, dass die durchschnittliche Arbeitsbelastung der Teilnehmer zunahm, je weiter sie von der Low-Fi- zur High-Fidelity-Version kamen. Darüber hinaus zeigen die Ergebnisse eine Zunahme der Frustration, der zeitlichen Anforderungen, der Anstrengung und im Gegensatz dazu eine Abnahme der subjektiven Wahrnehmung der erreichten Leistung mit jeder Wiedergabetreue [Su19b].

5 RQ2: Wie können wir das Software-Prototyping technologisch unterstützen?

In den letzten 25 Jahren wurden zahlreiche akademische und kommerzielle Tools eingeführt, um UI/UX-Designer beim UI-Prototyping zu unterstützen. Wir begannen unsere Analyse mit einer Untersuchung bestehender Prototyping-Tools (15 akademische, 140 kommerzielle) und dem Ausmaß der technischen Unterstützung, die sie für den Software-Prototyping-Prozess bieten. Unsere Literaturrecherche ergab, dass die vorhandenen Prototyping-Tools jede Fidelity als einen eigenständigen Schritt des UI-Prototyping behandeln. Das haben wir beobachtet:

Einige Zeichen-Tools unterstützen nur das Skizzieren, ohne weitere Unterstützung für das Prototyping zu bieten. Die meisten Tools beginnen den Entwurfsprozess mit dem me-fi und unterstützen nicht das Skizzieren von lo-fi oder sogar das Erstellen von Interaktionen. Einige wenige Tools unterstützen teilweise sowohl Lo-Fi als auch Me-Fi, erzeugen aber entweder CSS-Code-Schnipsel oder unterstützen Hi-Fi überhaupt nicht. Die meisten Tools bieten keine Möglichkeit zur Vorschau des UI-Designs

Intelligente Tools bieten keine Flexibilität bei der Skizzenerkennung Prozess. Im Wesentlichen kann der Benutzer nicht wählen, wann die Erkennung der Benutzeroberfläche stattfindet oder die Erkennungsergebnisse ändern. Sobald also eine Benutzeroberflächenskizze erkannt wurde, ist es nicht möglich, zum vorherigen Zustand zurückzukehren. Intelligente Tools, die UI-Skizzen erkennen, produzieren zwar einen gewissen Front-End-Code, aber sie unterstützen nicht die Verschönerung des UI-Designs und die Definition des Verhaltens.

Außerdem berücksichtigen sie nicht die Präferenzen des Designers, da sie ihre Standardkonfigurationen vorgeben.

Zusammenfassend lässt sich feststellen, dass trotz zahlreicher akademischer und kommerzieller Prototyping Prototyping-Tools, die direkt oder indirekt einen oder mehrere Aspekte des UI-Prototyping unterstützen, ein Mangel an Forschung zur Bereitstellung technologischer Lösungen besteht, um die Probleme zu lösen, mit denen Designern während des Prototyping-Prozesses in der Praxis konfrontiert sind.

Zu diesem Zweck haben wir den Prototyping-Prozess unter den folgenden Aspekten analysiert:

Traditionelles Prototyping

Eine formative Nutzerstudie mit halbstrukturierten Interviews mit 45 UI/UX-Designern hatte das Ziel, die gängigen Praktiken, Tools und Strategien zu verstehen, die UI/UX-Designer beim UI-Prototyping einsetzen. Wir waren auch daran interessiert, die Entwicklung, Interkonnektivität und Interdependenz aller drei Prototyping-Fidelitys in der Praxis zu untersuchen.

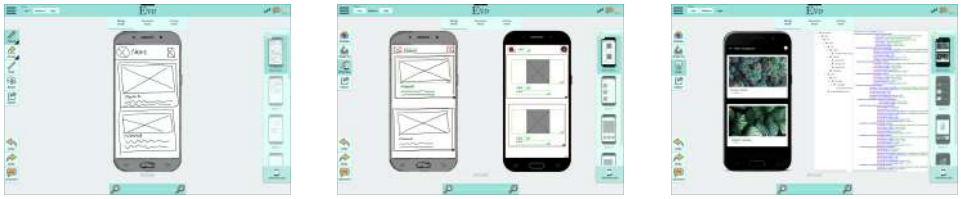
Insgesamt berichteten unsere Teilnehmer, dass das UI-Prototyping zahlreiche Iterationen des UI-Designs beinhaltet. Der problematische Teil sind nicht die Wiederholungen, sondern die *Überarbeitung* des Textes. Sie berichteten, dass sie das Design jedes Mal von Grund auf neu entwerfen mussten, wenn sie eine Fidelity in eine andere umwandelten. Sie empfanden dies als frustrierend, da es ihre *Gesamtarbeitsbelastung* erhöht.

Um dieses Problem anzugehen, untersuchten wir die vorhandene Toolunterstützung für die halbautomatische Umwandlung von LoFi in Me-Fi und dann in Hi-Fi-Prototypen unter Verwendung von Mustererkennung und Deep Learning für die Objekterkennung. Wir haben festgestellt, dass diese Projekte keine Flexibilität bei der Skizzenerkennung zulassen. Im Wesentlichen kann der Benutzer nicht wählen, wann die UI-Erkennung stattfindet oder die Erkennungsergebnisse ändern. Sobald eine Benutzeroberflächenskizze erkannt wurde, ist es also nicht möglich, zum vorherigen Zustand zurückzukehren.

Um diese Forschungslücke zu schließen, konzipierten und entwickelten wir Eve[4] [Su19b] (Abb. 1), eine hochinnovative Prototyping-Workbench, die den Benutzern eine *Canvas* zur Verfügung stellt, um ihr Konzept als Low-Fidelity-Prototyp zu skizzieren. Aus halbstrukturierten Interviews mit 18 UI/UX-Designern geht hervor, dass 88% der Befragten dazu neigen, den UI-Designprozess mit dem Skizzieren ihrer Ideen als Lo-Fi-Prototyp zu beginnen.

Im Hintergrund erkennt der *UI Element Detector: MetaMorph*[5] [Pa20] die skizzierten UI-Elemente mithilfe von Deep Neural Networks (84,9% mAP[6], 72,7% AR[7]). Das MetaMorph-

[4] `https://designwitheve.com/`
[5] `https://metamorph.designwitheve.com/`
[6] mAP bezieht sich auf die mittlere durchschnittliche Präzision (mean Average Precision)
[7] AR bezieht sich auf den durchschnittlichen Rückruf (Average Recall)



(a) Lo-Fi (b) Me-Fi (c) Hi-Fi

Abb. 1: Eve: eine skizzenbasierte Prototyping-Workbench

Objekterkennungsmodell (RetinaNet) wurde anhand unseres UISketch-Datensatzes trainiert, der 5.906 UI-Element-Skizzen und 125.000 synthetisch generierte Lo-Fi-Skizzen enthält [PSJ21, PSJ20]. Anhand der von MetaMorph bereitgestellten Informationen erstellt der *UI Element Generator* die entsprechenden UI-Elemente als Me-Fi. Schließlich wandelt der *Code Generator* Me-Fi in Hi-Fi als ausführbaren Code um. Wir haben Eve unter Verwendung von System Usability Scale (SUS) mit 15 UI/UX-Designern evaluiert; die Ergebnisse zeigen eine ausgezeichnete Benutzerfreundlichkeit und eine hohe Lernfähigkeit (SUS: 89,5).

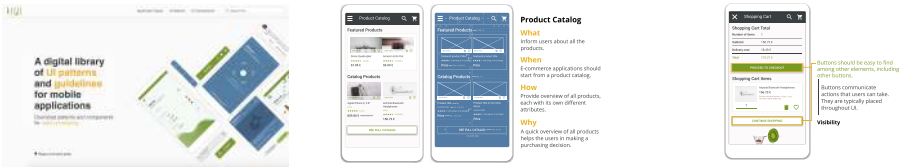
Rapid Prototyping

In der agilen Entwicklung führen Lean UX-Designer Rapid Prototyping durch, um schnelle Releases zu gewährleisten. Wir befragten 15 Lean UX-Designer und untersuchten Rapid Prototypen, um ihre Arbeitsabläufe beim Rapid Prototyping zu verstehen. Die Teilnehmer berichteten von Kompromissen bei der Qualität des UI-Designs aufgrund von knappen Fristen. Sie berichteten auch, dass die Entwickler aufgrund mangelnder Kenntnisse im Bereich des UI-Designs nicht in der Lage sind, die gleiche Qualität des UI-Designs mit Front-End-Code zu erreichen. Als problematisch erwies sich auch, dass das Wissen über UI-Design auf zahlreiche Quellen wie Websites und Bücher verstreut ist [Su19a].

Um diese Probleme zu lösen, schlagen wir einen UI-Design-Pattern-gesteuerten Ansatz für Rapid Prototyping vor. Um diesen Ansatz zu realisieren, wurde mit Kiwi[8]: eine webbasierte Bibliothek designt und entwickelt, die darauf abzielt, UI-Design-Wissen in Form von UI-Design-Patterns und -Richtlinien zu konsolidieren [Su19a] (Abb. 2). Wir gehen das Problem des verstreuten UI-Design-Wissens an, indem wir 108 UI-Design-Pattern aus verschiedenen Pattern-Sprachen von 4 Websites, Pattern-Sammlungen in 6 Büchern und 5 Pattern-Sammlungen im Internet zusammenführen. Zusätzlich haben wir 596 UI-Richtlinien aus 7 UI-Design-Büchern, 4 Forschungsarbeiten und 4 Web-Ressourcen gesammelt und 4 Leitfaden-Sammlungen für Web-Inhalte.

Jedes UI-Design-Pattern besteht aus einer Problemstellung (was), dem Kontext (wann), der Begründung (warum) und einem Lösungsvorschlag (wie). Zusätzlich bietet Kiwi herunterladbare GUI-Beispiele, UI-Layout-Entwürfe und Front-End-Code für jedes Muster.

[8] `https://designwithkiwi.com/`



(a) Kiwi-Bibliothek (b) UI-Design Pattern (c) UI-Design Richtlinien

Abb. 2: Kiwi: eine webbasierte Bibliothek mit UI-Design-Patterns und Richtlinien

Zusätzlich zu den UI Design Patterns enthält Kiwi UI Design Richtlinien aus verschiedenen Quellen, einschließlich akademischer Forschung und Industriestandards. Die Usability-Bewertung von Kiwi (SUS = 77,6) mit 21 schlanken UX-Designern zeigt eine gute Benutzerfreundlichkeit.

Prototyping für Barrierefreiheit

Barrierefreiheit bei der Gestaltung von Benutzeroberflächen führt zu einer befriedigenderen Erfahrung für alle Endbenutzer, unabhängig von ihren Fähigkeiten [Ca18]. Wir untersuchten, die Arbeitsabläufe verschiedener UI/UX-Designer, um UI-Designs zu erstellen, die für Benutzer mit visuellen, auditiven, kognitiven, sprachlichen und motorischen Behinderungen zugänglich sind. Dazu befragten wir 30 UI/UX-Designer, führten 21 Folgeinterviews durch und analysierten 32 Dokumente zur Benutzerprofilerstellung und zum UI-Design. Nach einer sorgfältigen Analyse haben wir die folgenden Probleme identifiziert: eingeschränkter Zugang zur Zielgruppe, Unsicherheit bezüglich der Priorität verschiedener Aspekte der Nutzerdaten, Unkenntnis der Richtlinien für barrierefreies UI-Design, Zeitmangel, der dazu führt, dass die Zugänglichkeit des UI-Designs vernachlässigt wird.

(a) Personify Bibliothek (b) Alle Personas (c) Persona's graph

Abb. 3: Personify: eine webbasierte Bibliothek von UI-Designrichtlinien für barrierefreie UIs

Um diese Probleme anzugehen, stellen wir Personify[9] vor, eine Bibliothek für UI-Design-Richtlinien, die bereits existierende UI-Designrichtlinien für Barrierefreiheit [Ca18] in Bezug auf Personas (Abb. 3) grafisch organisiert. Diese Personas stellen fiktive Charaktere mit visuellen, auditiven, kognitiven, sprachlichen und motorischen Behinderungen dar. Mit der Einführung dieser Bibliothek wollen wir Richtlinien für Barrierefreiheit mit den entsprechenden Personas verknüpfen, um die Auffindbarkeit, Auffindbarkeit und Nutzbarkeit von UI-Designrichtlinien für Barrierefreiheit zu verbessern. Personify unterstützt

[9] `https://designwithpersonify.com/`

so UX-Designer bei der Verwendung von UI-Designrichtlinien für die Erstellung barrierefreier UI-Designs. Die Bewertung von Personify mit 16 UI/UX-Designern zeigt eine überdurchschnittliche Benutzerfreundlichkeit (SUS = 76,4).

6 RQ3: Wie wirkt sich diese technologische Unterstützung für das Software-Prototyping auf die Arbeitsbelastung aus?

Neben der Toolunterstützung sollte die Arbeit auch (i) zum Verständnis der subjektiven Arbeitsbelastung von UI/UX-Designern während des Software-Prototypings beitragen und (ii) einen ersten Versuch zur Bewertung der Auswirkungen des Einsatzes technologischer Unterstützung auf die Arbeitsbelastung beim Software-Prototyping darstellen.

Es gibt vier Möglichkeiten, die Arbeitsbelastung zu messen: leistungsbezogen, indirekt, subjektiv und physiologisch. Diese Messungen haben jedoch ihre Grenzen. In Anbetracht des kreativen und künstlerischen Charakters des Software-Prototyping hielten wir leistungsbezogene, physiologische und indirekte Messverfahren nicht für geeignet, um die Arbeitsbelastung zu bewerten. Stattdessen wählten wir die subjektive numerische Messtechnik für unsere Forschung. Genauer gesagt, haben wir den NASA - Task load index (NASA-TLX) [Go10] verwendet.

Der NASA-TLX wurde entwickelt, um verschiedene Schwierigkeiten zu mildern, die durch die unterschiedlichen Definitionen von Arbeitsbelastung bei verschiedenen Personen verursacht werden. Anstatt eine einzige Skala zur Bewertung der Arbeitsbelastung zu verwenden, werden sechs verschiedene Unterskalen eingesetzt. Zu diesen Unterskalen gehören Physische Anforderung, Mentale Anforderung, Zeitliche Anforderung, Leistung, Anstrengung und Frustration. Folglich tragen diese sechs Unterskalen dazu bei, sechs verschiedene Aspekte zur Definition der Arbeitsbelastung zu berücksichtigen.

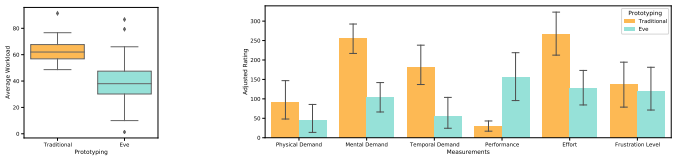
Der Hauptvorteil dieser Technik liegt darin, dass sie unterschiedliche Wahrnehmungen der Arbeitsbelastung berücksichtigt und Vorurteile hinsichtlich der Auswirkungen von unzureichender Leistung auf die Arbeitsbelastung ausräumt. Der größte Nachteil dieser Technik ist jedoch, dass sie zeitaufwändig ist und das Problem der Skalenbelastung mit sich bringt.

Unter Verwendung von NASA-TLX haben wir die Auswirkungen der von uns vorgeschlagenen technologischen Unterstützung auf die Arbeitsbelastung beim Software-Prototyping in den folgenden drei Aspekten bewertet:

Traditionelles Prototyping mit Eve

Unsere Arbeitsbelastung-Analyse (Abb. 4) zeigt, dass im Gegensatz zum traditionellen Prototyping-Ansatz die umfassende Unterstützung von Eve den Wechsel zwischen verschiedenen Prototyping-Tools während der Entwicklung von Lo-Fi, Me-Fi und Hi-Fi überflüssig macht. Folglich sinkt die subjektive Arbeitsbelastung von UI/UX-Designern, die Eve verwenden, erheblich. Auch die mentalen und zeitlichen Anforderungen sowie der

Arbeitsaufwand für UI/UX-Designer, die Eve verwenden, sind deutlich geringer. Im Vergleich zum traditionellen Ansatz stieg die wahrgenommene Gesamtleistung mit Eve um das Fünffache.

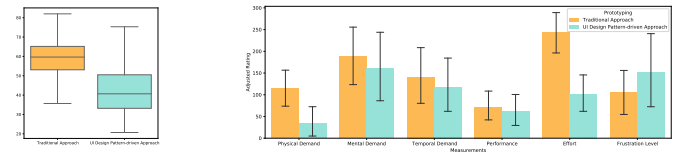


(a) Arbeitsbelastung (b) Subskalenvergleich

Abb. 4: Vergleich der durchschnittlichen Arbeitsbelastung beim UI-Prototyping mit dem traditionellen Ansatz und Eve.

Rapid Prototyping mit Kiwi

Hinsichtlich der Arbeitsbelastung zeigen unsere Ergebnisse, dass die subjektive Arbeitsbelastung von UI/UX-Designern, die den mustergesteuerten Ansatz mit Kiwi verwenden, signifikant geringer ist als die Arbeitsbelastung, die sie mit dem traditionellen Ansatz des Rapid Prototyping (Abb. 5) erfahren. Insbesondere ist eine signifikante Verringerung des physischen Bedarfs und des Aufwands beim Rapid Prototyping bei Verwendung des mustergesteuerten Ansatzes festzustellen. Es gibt jedoch keinen signifikanten Unterschied in der subjektiven Arbeitsbelastung bei der Verwendung von UI-Design-Pattern-Bibliotheken mit und ohne Pattern-Standard [SHJ20].

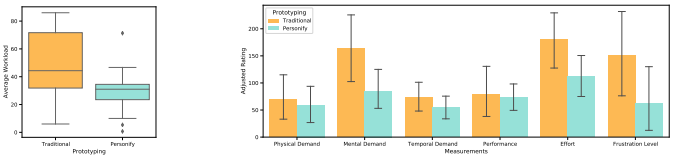


(a) Arbeitsbelastung (b) Subskalenvergleich

Abb. 5: Vergleich des durchschnittlichen Arbeitsaufwands beim Rapid Prototyping mit einem traditionellen und einem UI-Design-Pattern-gesteuerten Ansatz.

Prototyping für Barrierefreiheit mit Personify

Unsere Ergebnisse zeigen, dass die subjektive Arbeitsbelastung von UI/UX-Designern bei der Verwendung des Personify-Ansatzes deutlich geringer ist als bei der Verwendung des traditionellen Ansatzes der Prototypenerstellung für Barrierefreiheit (Abb. 6). Insbesondere sind die mentalen Anforderungen und der Aufwand für das Prototyping barrierefreier Benutzeroberflächen bei der Verwendung von Personify deutlich geringer.



(a) Arbeitsbelastung (b) Subskalenvergleich

Abb. 6: Vergleich des durchschnittlichen Arbeitsaufwands beim barrierefreien UI-Prototyping mit herkömmlichem und Personify.

7 Vorteile & Abschluss

Ziel dieser Untersuchung ist es, den kreativen Prozess des Software-Prototyping aus der Perspektive der Arbeitsbelastung zu analysieren. Wir beginnen mit der Durchführung einer Schmerzpunktanalyse mit UI/UX-Designern für traditionelles Prototyping, Rapid Prototyping und Prototyping für Barrierefreiheit. Wir gehen das Problem der Nacharbeit und der Zeitbeschränkung an, indem wir Deep Learning einsetzen, um den Fidelity- Transformationsprozess zu automatisieren. Darüber hinaus befasst sich der auf UI-Design-Patterns basierende Ansatz mit diesem Problem, indem er vorgefertigte Lösungen für sich wiederholende Probleme bietet.

Diese Forschung zielt darauf ab, Designern umfassende Lösungen zur Verfügung zu stellen, die den gesamten Prozess des Prototyping unterstützen. Wir stellen UI-Design-Wissen in einer einheitlichen Bibliothek bereit, um das Problem des verstreuten Wissens zu lösen. Wir stellen die Endbenutzer als Personas dar, um ihre Sichtbarkeit zu erhöhen und es den Designern zu erleichtern, sich in die Zielbenutzer einzufühlen. Schließlich wollen wir die Designer bei der Kommunikation des Designs mit den Entwicklern in Form von Entwürfen und Front-End-Code unterstützen.

Diese Arbeit zielt darauf ab, frühere Arbeiten zum UI-Prototyping zu erweitern. Sie ist allgemein anwendbar, um die Auswirkungen der Verwendung von Deep Learning, UI-Design-Patterns und Personas auf den Arbeitsaufwand des UI-Prototyping zu verstehen. Für die Zukunft planen wir, Eve zu verbessern, indem wir die Liste der erkannten UI-Elemente erweitern und die Genauigkeit und Präzision der Erkennung von UI-Elementen verbessern. Außerdem wollen wir weitere Patterns, Richtlinien und Personas in unsere Bibliotheken aufnehmen und sie weiter ausbauen durch die Integration anderer Plattformen, z. B. Web- und Smartwatch-Anwendungen.

Literaturverzeichnis

[Ca17] Camburn, Bradley; Viswanathan, Vimal; Linsey, Julie; Anderson, David; Jensen, Daniel; Crawford, Richard; Otto, Kevin; Wood, Kristin: Design prototyping methods: state of the art in strategies, techniques, and guidelines. Design Science, 3, 2017.

[Ca18] Caldwell, Ben; Reid, Loretta Guarino; Vanderheiden, Gregg; Chisholm, Wendy; Slatin, John; White, Jason: Web content accessibility guidelines (WCAG) 2.1. WWW Consortium (W3C), Juni 2018.

[Go10] Gore, Brian: Measuring and Evaluating Workload: A Primer. NASA Technical Memorandum, (July):35, 2010.

[IS10] ISO-9241-210: 9241-210: 2010. Ergonomics of human system interaction-Part 210: Human-centred design for interactive systems (formerly known as 13407). International Standardization Organization (ISO). Switzerland, 2010.

[Pa20] Pandian, Vinoth Pandian Sermuga; Suleri, Sarah; Beecks, Christian; Jarke, Matthias: MetaMorph: AI Assistance to Transform Lo-Fi Sketches to Higher Fidelities. In: Proceedings of the 32nd Australian Conference on HCI. ozCHI'20, Association for Computing Machinery, New York, NY, USA, 2020.

[PSJ20] Pandian, Vinoth Pandian Sermuga; Suleri, Sarah; Jarke, Matthias: Syn: Synthetic Dataset for Training UI Element Detector From Lo-Fi Sketches. In: Proceedings of the 25th International Conference on Intelligent User Interfaces Companion. IUI '20, Association for Computing Machinery, New York, NY, USA, S. 79–80, 2020.

[PSJ21] Pandian, Vinoth Pandian Sermuga; Suleri, Sarah; Jarke, Matthias: UISketch: A Large-Scale Dataset of UI Element Sketches. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. CHI '21, Association for Computing Machinery, New York, NY, USA, 2021.

[SHJ20] Suleri, Sarah; Hajimiri, Yeganeh; Jarke, Matthias: Impact of using UI Design Patterns on the Workload of Rapid Prototyping of Smartphone Applications: An Experimental Study. In: Proceedings of the 22nd International Conference on Human-Computer Interaction with Mobile Devices and Services. MobileHCI '20, Association for Computing Machinery, New York, NY, USA, 2020.

[Su19a] Suleri, Sarah; Kipi, Nilda; Tran, Linh Chi; Jarke, Matthias: UI Design Pattern-Driven Rapid Prototyping for Agile Development of Mobile Applications. In: Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services. MobileHCI '19, Association for Computing Machinery, New York, NY, USA, 2019.

[Su19b] Suleri, Sarah; Pandian, Vinoth Pandian Sermuga; Shishkovets, Svetlana; Jarke, Matthias: Eve: A Sketch-based Software Prototyping Workbench. In: Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems. CHI EA '19, ACM, New York, NY, USA, S. Lbw1410:1—-lbw1410:6, 2019.

[Su21] Suleri, Sarah: Impact of technological support on the workload of software prototyping. Dissertation, RWTH Aachen University, Aachen, 2021. Veröffentlicht auf dem Publikationsserver der RWTH Aachen University; Dissertation, RWTH Aachen University, 2021.

Sarah Suleri zog 2014 von Pakistan nach Deutschland, um ihren Master in Medieninformatik zu machen. Nach ihrem Abschluss kam sie als UX Research Associate zu Fraunhofer und arbeitete an verschiedenen H2020-Forschungsprojekten. Neben ihrer Vollzeitstelle bei Fraunhofer forschte sie im Rahmen ihrer Promotion unter der Leitung von Prof. Matthias Jarke. Während ihrer Promotion hat sie auf zahlreichen Tier-1-Konferenzen wie CHI, IUI, DIS und MobileHCI veröffentlicht. Sie beendete ihre Doktorarbeit in weniger als vier Jahren und schloss sie mit summa cum laude (Auszeichnung) ab. In Anerkennung ihrer akademischen Leistungen wurde sie mit der Borchers-Plakette und der Springorum-Gedenkmünze 2021 ausgezeichnet. Sarah verfolgt weiterhin ihre Leidenschaft für UX-Design in der Industrie.

Die Wirksamkeit von Graphalgorithmen: Effiziente Algorithmen für die formale Verifikation[1]

Alexander Svozil[2]

Abstract: In der formalen Verifikation versucht man Fehler von Systemen automatisch zu finden. Ein *Modellprüfer* kontrolliert ob ein gegebenes *Modell* eines Systems eine *Anforderung* erfüllt. In der *reaktiven Synthese* wird mit einer gegebenen Anforderung ein korrektes *reaktives System* erzeugt. Wir verbinden das Gebiet der modernen theoretischen Graphalgorithmen mit dem Gebiet der formalen Verifikation indem wir schnellere Algorithmen für algorithmische Probleme in der Modellprüfung und der reaktiven Synthese in der Dissertation vorstellen. Neben gewohnten "expliziten" Algorithmen stellen wir *symbolische Algorithmen* vor – Symbolische Algorithmen erlauben zwar nur einen limitierten Zugang zur Eingabe, ermöglichen aber eine effizientere Speicherrepräsentation.

1 Einführung und Motivation

In den letzten Jahrzehnten haben Computersysteme unsere Welt in vielen Bereichen massiv bereichert und Probleme vereinfacht – man denke zum Beispiel an Navigation am Mobiltelefon oder endlose streambare Videoinhalte. Weil diese Systeme von Menschen erstellt werden, sind sie oft mit Programmierfehlern versehen die schwer zu erkennen sind; Übersehene Fallunterscheidungen, falsch geschriebene Variablennamen oder nicht initialisierte Pointer sind nur eine Bruchteil der möglichen Fehler die passieren können wenn wir Systeme erstellen. Fehler in sicherheitskritischen Systemen haben teilweise katastrophale Auswirkungen: Zum Beispiel scheiterte der Start der Ariane 5 Rakete an einem Fehler bei einer Typumwandlung von einer 64-Bit auf eine 16-Bit Variable [Be01].

In der Praxis schreibt man *Tests* um Fehler vorzubeugen. Ein Test ist erfolgreich wenn ein System mit gegebenen Input den gewünschten Output erzeugt. Leider sind Tests im Allgemeinen kein Zertifikat für ein fehlerfreies System. Besonders in parallelen Systemen reicht ein Durchgang solcher Tests nicht: Es kann wegen der gleichzeitigen Ausführung passieren, dass ein Test in einem Durchgang erfolgreich ist und im nächsten Durchgang scheitert. In der formalen Verifikation *beweist* man das ein System korrekt ist, zum Beispiel, dass es eine gewisse Eigenschaft erfüllt. Wegen der Unentscheidbarkeit des Halteproblems [Tu37] und Rice's Theorem [Ri53] kann man im Allgemeinen nicht beweisen das ein System korrekt ist. Selbst wenn wir uns auf Systeme mit einer endlichen Anzahl an

[1] Englischer Titel der Dissertation [Sv22]: "Leveraging the Power of Graph Algorithms: Efficient Algorithms for Computer-Aided Verification"

[2] Universität Wien, Theory and Application of Algorithms, Währinger Straße 29, 1090 Wien, Österreich alexander.svozil@gmail.com

Zuständen beschränken – hier ist das Problem entscheidbar – sind viele interessante Fragen der formalen Verifikation wegen Ergebnissen aus der Komplexitätstheorie schwer zu lösen. Nichtsdestotrotz erfanden Clarke, Emerson, Sifakis und andere einen praktischen Ansatz den Sie "Modellprüfung" [CES09] nennen. In der Modellprüfung abstrahiert man das System mit einem Modell und überprüft ob das Modell sich wie gewünscht verhält. Gegeben ein System und eine Spezifikation – eine Vorschrift wie ein Programm sich verhalten soll – verwandelt man das System zuerst in einen *endlichen Zustandsgraphen*; das Modell. Dann übersetzt man die Spezifikation in *Zielvorgaben*. Der Input des Modellprüfers ist das Modell und die Zielvorgaben. Der Modellprüfer meldet ob der endliche Zustandsgraph die gewünschten Zielvorgaben erfüllt oder ob es ein Gegenbeispiel gibt. Wenn es ein Gegenbeispiel gibt hat entweder das System einen Fehler oder das Modell repräsentiert das System nicht ausreichend.

Reaktive Systeme interagieren laufend mit der Umgebung – andere parallele Prozesse, Benutzereingaben, etc. Ein *Zustand* in einem reaktiven System ist eine Variablenzuweisung zu einem gegebenen Zeitpunkt. Die Semaphore ist ein Beispiel für ein reaktives System: Wenn mehrere parallele Prozesse ein kritisches Codestück ausführen wollen regelt sie den Zugriff darauf. Für ein reaktives System sind Endzustände meist unerwünscht – die Semaphore wäre in diesem Fall im Deadlock – und deshalb hat jeder Zustand einen Nachfolger. Ein Flugzeugkontrollsystem oder ein Flugzeug sind fortgeschrittene reaktive Systeme.

Modelle. Das Standardmodell ist ein endlicher gerichteter Graph. Die Knoten des Graphen repräsentieren die Zustände des Systems und die Kanten repräsentieren die Übergänge zwischen Zuständen im System – zum Beispiel wenn eine Variable inkrementiert wird. Einen Systemdurchlauf modelliert man als *Spiel*; ein unendlicher Pfad der an dem Knoten startet der den initialen Zustand des Systems repräsentiert. Für manche Systeme ist der endliche gerichteter Graph ungenügend um das gewünschte System zu repräsentieren; man benutzt deshalb stattdessen oft *Spielgraphen* und *Markow-Entscheidungsprozesse(MEPs)*.

Spielgraphen werden in der *Synthese* verwendet; dort erstellt man ein reaktives System in einem zugbasiertem Spiel [Bü62; Ch62]. In Spielgraphen gibt es zwei Spieler: Spieler 1 repräsentiert das System und Spieler 2 repräsentiert die Umgebung. Die Knoten werden in Spieler-1 Knoten und Spieler-2 Knoten aufgeteilt. Zu Beginn des Spiels ist ein Spielstein auf dem initialen Knoten. Wenn der Spielstein auf einem Spieler-1 Knoten ist, darf Spieler-1 ihn entlang der ausgehenden Kanten zum nächsten Knoten bewegen. Analog darf Spieler-2 den Spielstein bewegen wenn er sich auf einem Spieler-2 Knoten befindet.

Markow-Entscheidungsprozesse (MEPs) modellieren Systeme die mit einer nicht-deterministischen oder unsicheren Komponente interagieren [Va85]: In MEPs gibt es Spieler-1 Knoten und Zufallsknoten. Am Anfang des Spiels ist ein Spielstein auf dem initialen Zustand. Wenn sich der Spielstein auf einem Spieler-1 Knoten befindet, bewegt das System den Spielstein entlang der ausgehenden Kanten. Wenn der Spielstein auf einem

Zufallsknoten ist, wird der nächste Knoten durch eine Wahrscheinlichkeitsverteilung über die ausgehenden Kanten entschieden.

Spezifikationen. Eine Spezifikation bestimmt die gewünschten Eigenschaften bei der Modellprüfung und die reaktive Synthese. *Sicherheitszielvorgaben* sind ein einfaches Beispiel für eine Eigenschaft eines Systems – bei einer Sicherheitszielvorgabe muss ein System sicherstellen, dass eine Menge von unerwünschten Zuständen nicht im Spiel vorkommen. Formell sind *Zielvorgaben* immer eine Menge von Spielen, das heißt eine Menge von erlaubten Systemdurchläufen. In der Dissertation betrachten wir folgende Zielvorgaben im Detail:

Erreichbarkeits- und Sicherheitszielvorgaben. Gegeben ist eine Menge von "guten" Knoten. Die *Erreichbarkeitszielvorgabe* ist die Menge von Spielen die einen guten Knoten beinhalten. Dual dazu, gegeben eine Menge aus "sicheren" Knoten, besteht die *Sicherheitszielvorgabe* aus der Menge von Spielen die nur die sicheren Knoten besucht.

Sequenzielle Erreichbarkeitszielvorgaben. Gegeben k Mengen von Knoten besteht die *sequenzielle Erreichbarkeitszielvorgabe* aus der Menge von Spielen die zuerst einen Knoten aus der ersten Menge erreichen, dann einen Knoten aus der zweiten und so weiter, bis das Spiel einen Knoten aus der $k$ten Menge erreicht.

Büchi- und coBüchizielvorgaben. Die *Büchizielvorgaben* beschreiben, gegeben eine Menge aus "Büchi Knoten", die Menge von Spielen die einen beliebigen Büchi-Knoten unendlich oft besucht. Dual dazu, gegeben eine Menge von "coBüchi-Knoten" beschreiben die *coBüchizielvorgaben* die Menge von Spielen die nur die coBüchi-Knoten unendlich oft besuchen.

Beschränkte Büchi- und beschränkte coBüchizielvorgaben. Beschränkte Büchizielvorgaben erweitern die oben genannten Büchizielvorgaben: Gegeben eine natürliche Zahl d und eine Menge von Büchi-Knoten muss ein Spiel nach endlich vielen Schritten – um in den beschränkten Büchizielvorgaben zu sein – jede höchstens d Schritte einen Büchi-Knoten besuchen. Dual dazu muss ein Spiel in den beschränkten coBüchizielvorgaben unendlich oft d coBüchi-Knoten besuchen.

Paritätszielvorgaben. Bei *Paritätszielvorgaben* wird jedem Knoten eine natürliche Zahl – eine *Priorität* – zugewiesen. Die Paritätszielvorgabe beinhaltet Spiele wo die unendlich oft vorkommenden Knoten mit der niedrigsten Priorität *gerade* sind.

Streettzielvorgaben. Streettzielvorgaben haben eine Menge von Bedingungen und eine Menge von dazugehörigen Genehmigungen. Eine Bedingung – für eine Genehmigung gilt dasselbe – ist eine Menge von Knoten im Modell. Eine Streettzielvorgabe beinhaltet ein Spiel wenn es für jeden unendlich oft besuchten Knoten in einer Bedingung einen Knoten in der dazugehörigen Genehmigung besucht.

Mittelwertzielvorgaben. Bei *Mittelwertszielvorgaben* wird jeder Kante im Modell eine Belohnung zugeordnet. Jedes Mal wenn der Stein von einem Knoten zum nächsten bewegt wird, erhält Spieler 1 die Belohnung der Kante. Der *Mittelwert eines Spiels* ist der Limes des Durchschnitts der erhaltenen Belohnungen. *Mittelwertszielvorgaben* haben zusätzlich einen Grenzwert gegeben; ein Spiel ist in den Mittelwertszielvorgaben wenn der Mittelwert des Spiels über dem Grenzwert liegt.

Kombinierte Mittelwerts- und Paritätszielvorgaben. Ein Spiel ist in den *kombinierten Mittelwerts- und Paritätszielvorgaben* wenn es in den gegebenen Paritätszielvorgaben und in den Mittelwertszielvorgaben ist.

Algorithmische Fragen. Wir untersuchen die folgenden zwei algorithmischen Fragen:

Frage 1: Gegeben ein Modell, eine Zielvorgabe und den initialen Knoten im Modell, berechnen wir ob Spieler 1 die Nachfolger seiner Knoten so wählen kann, dass das resultierende Spiel in der Zielvorgabe ist. In Spielgraphen versucht Spieler 2 dieses Ziel zu verhindern. In MEPs kann es passieren, dass der Nichtdeterminismus die Aufgabe verhindert. Wenn Spieler 1 erzwingen kann das ein Spiel von dem initialen Knoten in der Zielvorgabe liegt, dann *gewinnt der Knoten für Spieler 1*. Die Menge aller Knoten, die für Spieler 1 gewinnen, nennt man *Gewinnmenge*.

Frage 2: Gegeben ein MEP, berechnen wir die maximalen Schluss-Komponenten Dekomposition (MSK). Intuitiv beschreibt eine MSK die maximale (unter Mengenvereinigung) Menge von Knoten in einem MEP für die Spieler 1 jeden anderen Knoten trotz der Zufallselemente in einem MEP im MSK besuchen kann. Deswegen verallgemeinern MSKs starke Zusammenhangskomponenten in Graphen. Die Berechnung von MSKs in einem MEP sind eine Schlüsselkomponente zur Berechnung von ω-regulären Zielvorgaben in MEPs [Ch07].

Symbolische Algorithmen und das Zustandsexplosionsproblem. Ein Zustand eines reaktiven Systems besteht aus einer Variablenzuweisung zu einem gewissen Zeitpunkt. Das führt zu einer riesigen Anzahl an Knoten in dem Modell weil die Anzahl der Zustände exponentiell mit der Anzahl der Variablen wächst: Ein Bit-Array mit 20 Einträgen und zwei Variablen mit Werten in $\{0,\ldots,9\}$ ergibt schon $2^{20} \cdot 10^2$ mögliche Zustände – oft passen Modelle deshalb nicht in den Arbeitsspeicher. Zur Lösung dieses Problems wurden Binäre Entscheidungsdiagramme (BEDs) erfunden: Hier werden Mengen von Zuständen und Zustandsübergänge implizit anstatt explizit in Form von BEDs dargestellt [Br92]. Wir betrachten das symbolische Modell der Berechnung, ein theoretisches Modell für Algorithmen mit BEDs, dass die implizite Darstellung der BEDs vernachlässigt. Ein symbolischer Algorithmus kann dieselben Operationen wie ein Algorithmus im RAM Modell verwenden – außer wenn er auf das Modell zugreift: Um auf den Eingabemodell zuzugreifen muss ein symbolischer Algorithmus eine der folgenden zwei Typen von *symbolischer Operationen* verwenden:

1. Schritt-Operationen $\mathrm{Pre}(\cdot)$ und $\mathrm{Post}(\cdot)$: Gegeben eine Menge von Knoten X, gibt die Vorgängerfunktion $\mathrm{Pre}(X)$ die Menge von Knoten zurück die eine Kante zu einem Knoten in X haben. Ähnlich gibt die Nachfolgerfunktion $\mathrm{Post}(X)$ die Menge von Knoten zurück, die eine Kante von einem Knoten in X haben.
2. Klassische Mengenoperationen. Klassische Mengenoperationen haben eine oder zwei Mengen von Knoten als Eingabe und führen auf diese zum Beispiel das Komplement, die Vereinigung oder den Schnitt aus.

Im symbolischen Modell ist *die Laufzeit* als die Anzahl der symbolischen Operationen definiert. Eine Speichereinheit im symbolischen Modell ist eine Menge: Es ist egal, wie groß die Menge ist weil BEDs die Menge implizit repräsentieren. Der *Speicherverbrauch eines symbolischen Algorithmus* ist die größte Anzahl gleichzeitig benutzter Mengen.

Moderne Graphalgorithmen und Techniken. Wir benutzen die Wirksamkeit von modernen Graphalgorithmen um schnellere Algorithmen für die algorithmischen Fragestellungen zu finden. Außerdem stellen wir negative Resultate in Form von "konditionalen unteren Schranken" vor, d.h., wir zeigen das ein besseres Ergebnis für ein algorithmisches Problem eine langstehende Laufzeitschranke für bekannte Probleme wie SAT durchbrechen würde. Die folgenden drei algorithmischen Konzepte sind Teil unserer Werkzeugkiste:

Dynamische Graphalgorithmen. Ein *dynamischer Graphalgorithmus* bewahrt eine Eigenschaft eines Graphen (zum Beispiel stark zusammenhängende Komponenten in einem Graphen) während Kanten im Graph hinzugefügt und entfernt werden. Diese Algorithmen sind in der Regel besser als die naive Neuberechnung der Eigenschaft nach jeder Entfernung und Einfügung einer Kante. Wir verwenden dynamische Graphalgorithmen als Routinen in unseren statischen Algorithmen, zum Beispiel, wenn Knoten wiederholt entfernt werden.

Hierarchische Graphzerlegung. Die hierarchischen Graphzerlegung ist eine Technik die ursprünglich für dynamische Graphen entwickelt wurde aber ein bahnbrechendes Ergebnis zeigte das man sie in Spielgraphen verwenden kann wenn wir wiederholt Mengen von Knoten löschen [CH14].

Konditionale Untere Schranken. Ähnlich zu gegenseitigen Reduktionen von NP-schweren Problem geben konditionale untere Schranken Garantien über die Laufzeit. Wenn eine konditionalen unteren Schranke gezeigt für ein Problem A gezeigt ist ergibt jede Verbesserung der Laufzeit für Problem A einen neuen Algorithmus für ein "beliebtes" Problem das seit Jahrzehnten keine bedeutend bessere Laufzeit mehr gefunden wurde. Beispiele für "beliebten" Problem sind All-Pair Shortest Path (APSP), 3-SUM und CNF-SAT.

2 Forschungsstand

In diesem Bereich beschreiben wir den derzeitigen Forschungsstand für die algorithmischen Fragen grob. Wir beschreiben die Laufzeit des Algorithmus für ein Modell mit n Knoten, und m Kanten. Bei Paritätszielvorgaben betrachten wir Modelle mit d Prioritäten; Bei Mittelwertszielvorgaben ist W das maximale Gewicht einer Kante. Bei Streett Zielvorgaben beschreibt b die Größe der Bedingungsmengen und der Genehmigungsmengen. Die $\widetilde{O}(\cdot)$ Notation versteckt poly-logarithmische Faktoren. Die "Vorher" Spalte von Tab. 1 und Tab. 2 fassen die früheren Laufzeiten der Probleme die wir verbessert haben zusammen Eine vollständige Ausführung ist in meiner Doktorarbeit.

MSK Dekomposition. Explizite Algorithmen können die MSK Dekomposition in $O(\min(m^{1.5}, n^2))$ bestimmen [CH14]. Der klassische symbolische Algorithmus benötigt $O(n)$ viele symbolische Berechnungen starker Zusammenhangskomponenten die in $O(n)$ symbolischen Operationen und $O(\log n)$ symbolischen Speicher durchgeführt werden können [Ch18a]. Der zweite Algorithmus benötigt $O(n\sqrt{m})$ symbolische operationen und $O(\sqrt{m})$ symbolischen Speicher [Ch18b].

Streettzielvorgaben. Der besten explizite Algorithmus berechnet Streettzielvorgaben in Zeit $O(m^{1.5}\sqrt{\log n})$ und $O(n^2)$ [Ch16].

Paritätszielvorgaben. Eine lange Liste von Arbeiten verbesserte schrittweise die Laufzeit zur Berechnung von Gewinnmengen für Paritätszielvorgaben – besonders hervorgehoben sei die Entdeckung eines quasi-polynomialen Algorithmus mit Laufzeit $O(n^{\log d+6})$ [Ca17]. Seitdem gibt es wiederum eine Reihe von anderen Algorithmen mit quasipolynomialer Laufzeit aber die faszinierende Frage, ob es einen Algorithmus mit polynomialer Laufzeit gibt bleibt offen. Mittlerweile wurde unser symbolischer Algorithmus noch etwas verbessert [JM20] aber vor unserem Algorthmus hatte der beste Algorithmus $\min(n^{O(\sqrt{n})}, O(n^{d/3+1}))$ symbolische Schritte mit $O(n)$ symbolischen Speicherverbrauch [Ch17].

Mittelwert-Paritätszielvorgaben. Der beste Algorithmus benötigt $O(dmn^{\lg(d/\lg n)+2.45}W)$ nachdem er unseren Algorithmus durch einen quasipolynomialen Algorithmus ersetzt hat [DJL18]. Vor unserem Algorithmus benötigte der beste Algorithmus zur Berechnung der Gewinnmenge die Laufzeit $O(n^{d+1})$ [Bo11].

3 Ergebnisse

In diesem Bereich fassen wir die Ergebnisse der Dissertation zusammen. Eine kompakte Auflistung der neuen Laufzeiten sind in Tab. 1 und Tab. 2.

Algorithmen für Spielgraphen mit Mittelwert Büchi Zielvorgaben und Mittelwert co-Büchi Zielvorgaben. Wir präsentieren Algorithmen zur Berechnung der Gewinnmengen

Problem	Modell	Vorher	Neu
Mittelwert-Büchi	**S**	$O(n^3mW)$	$\boldsymbol{O(nmW)}$
Mittelwert-Parität	**S**	$O(n^{d+1}mW)$	$\boldsymbol{O(n^{d-1}mW)}$
MSK-Dekomposition	**M**	$O(\min(m^{1.5}, n^2))$	$\boldsymbol{\widetilde{O}(m)}$
Streett	**M**	$O(\min(m^{1.5}\sqrt{\log n}, n^2))$	$\boldsymbol{\widetilde{O}(m)}$
Beschränktes Büchi	**G**	$O(n^3)$	$\boldsymbol{O(n^{2.5}\log n)}$
Beschränktes Büchi	**S**	$O(n^4)$	$\boldsymbol{O(n^3)}$
Sequenzielle Erreichbarkeit	**M**	O(nm)	$\boldsymbol{O(m + \sum_{i=1}^{k} \lvert T_i \rvert)}$
Sequenzielle Erreichbarkeit	**S**	O(nm)	$\boldsymbol{\Omega(nm)}$
Abdeckungsproblem	**M,S**	O(nm)	$\boldsymbol{\Omega(nm)}$

Tab. 1: Explizite Algorithmen, **G**: Graphen, **M**: MEPs, **S**: Spielgraphen

		symb. Operationen		symb. Speicherplatz	
Problem	Modell	Vorher	Nacher	Vorher	Nacher
MSK-Dekomposition	**M**	$O(n\sqrt{m})$	$\boldsymbol{\widetilde{O}(n^{2-\epsilon})}$	$O(\sqrt{m})$	$\boldsymbol{\widetilde{O}(n^{\epsilon})}$
Parität	**M**	$O(n\sqrt{md})$	$\boldsymbol{\widetilde{O}(n^{2-\epsilon})}$	$O(\sqrt{m})$	$\boldsymbol{\widetilde{O}(n^{\epsilon})}$
Parität	**S**	$O(n^{d/3+1})$	$\boldsymbol{n^{O(d\log n))}}$	$O(n)$	$\boldsymbol{O(d\log n)}$

Tab. 2: Symbolische Algorithmen, **M**: MEPs, **S**: Spielgraphen

von Mittelwert Büchi Zielvorgaben und Mittelwert coBüchi Zielvorgaben. Diese neuen Algorithmen verbessern die Laufzeit von $O(n^3mW)$ zu $O(nmW)$. Außerdem präsentieren wir einen neuen Algorithmus zur Berechnung der Gewinnmenge von kombinierten Mittelwert Paritätsszielvorgaben der eine Laufzeit von $O(n^{d-1}mW)$ hat. Die wichtigste Entdeckung ist ein dynamischer Algorithmus der es erlaubt wiederholt eine Sequenz von Knotenmengen zu löschen ohne die Laufzeit zu erhöhen.

Algorithmen für die Berechnung der Gewinnmenge von Streett-Zielvorgaben in MEPs. Wir präsentieren einen Algorithmus in $\widetilde{O}(m+b)$ Zeit zur Berechnung der Gewinnmenge von Streett Zielvorgaben in Graphen und MEPs. Dazu präsentieren wir Algorithmen mit $\widetilde{O}(m)$ Laufzeit für (i) die Berechnung der MSK Dekomposition in MEPs, (ii) einen dynamischen Algorithmus welcher die MSK Dekomposition eines MEPS unter Kantenlöschungen berechnet und (iii) Berechnung der Gewinnmenge von Erreichbarkeitszielvorgaben in MEPs. Die wichtigste Komponente dieser Algorithmen ist ein spannendes Ergebnis für dynamische Algorithmen, dass die maximalen Zusammenhangskomponenten in $\widetilde{O}(m)$ Zeit berechnet [BPW19] und sich auf die behandelten Probleme anwenden lässt.

Algorithmen für beschränkte Büchi Zielvorgaben. Wir präsentieren die ersten subkubischen Algorithmen für die Berechnung der Gewinnmenge von eingeschränkten Büchi

Zielvorgaben in Graphen. Für Spielgraphen präesentieren wir einen neuen Algorithmus mit Laufzeit $O(n^2 d)$ indem wir die hierarchische Graphdekomposition benutzen. Die Schlüsselkomponente dieser Ergebnisse sind (1) die hierarchische Graphdekomposition und (2) ein randomisierter Algorithmus, der Knoten zufällig auswählt und ausnützt, dass diese Knotenmenge wahrscheinlich alle langen Pfade im Graphen abdeckt.

Algorithmen und untere Schranken für erweiterte Erreichbarkeits-Zielvorgaben. Wir betrachten Erreichbarkeitsprobleme mit k Mengen von Knoten in Spielgraphen und MEPs. Insbesondere präsentieren wir Algorithmen für die Berechnung von Gewinnmengen von sequenziellen Erreichbarkeitszielvorgaben und des Abdeckungsproblems – anstatt einer Erreichbarkeitszielvorgabe muessen hier k Erreichbarkeitszielvorgaben gleichzeitig erfüllt werden. Wir zeigen konditionale untere Schranken für sequenziellen Erreichbarkeitszielvorgaben in Spielgraphen. Für MEPs präsentieren wir einen subkubischen Algorithmus welcher konditionalen unteren Schranken ausschließt. Für das Abdeckungsproblem finden wir neue konditionale untere Schranken in Spielgraphen und MEPs.

Symbolische Algorithmen für Paritäts Zielvorgaben. Wir präsentieren den ersten symbolischen Algorithmus für Paritätszielvorgaben in Spielgraphen mit quasi-polynomialer Anzahl von symbolischen Operationen und $O(d \log n)$ symbolischen Speicher. Die beiden Ergebnisse gelingen durch die Übersetzung des quasi-polynomiellen Algorithmus [Ca17] in die Welt der symbolischen Algorithmen und durch eine geschickte Anwendung einer Datenstruktur die bereits in [Ch17] verwendet wurde.

Symbolische Algorithmen für die MSK Dekomposition und Paritätszielvorgaben in MEPs. Unser Hauptergebnis ist ein Algorithmus zur Berechnung der MSK Dekomposition mit einem symbolischen Zeit- und einem symbolischen Speicheraustausch. Unser Algorithmus braucht $\widetilde{O}(n^{2-\epsilon})$ symbolische Operationen und $\widetilde{O}(n^{\epsilon})$ symbolischen speicher für $0 < \epsilon \leq 1/2$. Das zweite Ergebnis ist ein Algorithmus, der die Gewinnmenge von Paritätszielvorgaben mit $\log d$ Berechnungen der MSK dekomposition und verbessert das Zeit-Speicherplatzprodukt von $\widetilde{O}(n^2 d)$ zu $\widetilde{O}(n^2)$. Das Ergebnis gelang primär durch die Erfindung eines schnellen symbolischen dynamischen Algorithmus, der es erlaubte die MSK Dekomposition unter Einfügung von Knoten effizient zu berechnen.

4 Schlussworte

In der Dissertation untersuchen wir zentralen Problem aus der Modellprüfung und der reaktiven Synthese, die wir dann mit modernen theoretischen Graphalgorithmen lösen. Wir schließen mit Ideen für zukünftige Arbeit ab: Die Entdeckung verbesserter theoretischer Algorithmen ist wichtig, aber wir müssen diese Algorithmen implementieren um einen

echten Nutzen zu ziehen; eine experimentelle Auswertung der Algorithmen würde zeigen was in der Praxis am schnellsten funktioniert. Vor kurzem gab es einen Durchbruch bei deterministischen dynamischen Algorithmen für viele zentrale Probleme in Graphen [BGS20]; es ist eine interessante offene Frage ob sich diese Ergebnisse auf die erwähnten Probleme in der formalen Verifikation übertragen lassen. Abschließend sei noch die anspruchsvolle offene Frage erwähnt, ob man die Gewinnmenge von Paritätszielvorgaben in Polynomialzeit berechenen kann.

Literatur

[Be01] Ben-Ari, M.: The bug that destroyed a rocket. ACM SIGCSE Bull. 33/2, S. 58–59, 2001, URL: https://doi.org/10.1145/571922.571958.

[BGS20] Bernstein, A.; Gutenberg, M. P.; Saranurak, T.: Deterministic Decremental Reachability, SCC, and Shortest Paths via Directed Expanders and Congestion Balancing. In: FOCS. IEEE, S. 1123–1134, 2020, URL: https://doi.org/10.1109/FOCS46700.2020.00108.

[Bo11] Bouyer, P.; Markey, N.; Olschewski, J.; Ummels, M.: Measuring Permissiveness in Parity Games: Mean-Payoff Parity Games Revisited. In: ATVA. LNCS 6996, Springer, S. 135–149, 2011.

[BPW19] Bernstein, A.; Probst, M.; Wulff-Nilsen, C.: Decremental Strongly-Connected Components and Single-Source Reachability in Near-Linear Time. In: STOC. S. 365–376, 2019.

[Br92] Bryant, R. E.: Symbolic Boolean Manipulation with Ordered Binary-decision Diagrams. ACM Comput. Surv. 24/3, S. 293–318, Sep. 1992, ISSN: 0360-0300.

[Bü62] Büchi, J. R.: On a decision method in restricted second-order arithmetic. In: Proceedings of the First International Congress on Logic, Methodology, and Philosophy of Science 1960. S. 1–11, 1962.

[Ca17] Calude, C. S.; Jain, S.; Khoussainov, B.; Li, W.; Stephan, F.: Deciding parity games in quasipolynomial time. In: STOC. S. 252–263, 2017.

[CES09] Clarke, E. M.; Emerson, E. A.; Sifakis, J.: Model checking: algorithmic verification and debugging. Commun. ACM 52/11, S. 74–84, 2009, URL: https://doi.org/10.1145/1592761.1592781.

[Ch07] Chatterjee, K.: Stochastic ω-Regular Games, Diss., UC Berkeley, 2007.

[CH14] Chatterjee, K.; Henzinger, M.: Efficient and Dynamic Algorithms for Alternating Büchi Games and Maximal End-Component Decomposition. J. ACM 61/3, 15:1–15:40, 2014, URL: https://doi.org/10.1145/2597631.

[Ch16] Chatterjee, K.; Dvořák, W.; Henzinger, M.; Loitzenbauer, V.: Model and Objective Separation with Conditional Lower Bounds: Disjunction is Harder Than Conjunction. In: LICS. ACM, S. 197–206, 2016, ISBN: 978-1-4503-4391-6.

[Ch17] Chatterjee, K.; Dvořák, W.; Henzinger, M.; Loitzenbauer, V.: Improved Set-Based Symbolic Algorithms for Parity Games. In: CSL. 18:1–18:21, 2017.

[Ch18a] Chatterjee, K.; Dvořák, W.; Henzinger, M.; Loitzenbauer, V.: Lower Bounds for Symbolic Computation on Graphs: Strongly Connected Components, Liveness, Safety, and Diameter. In: SODA. S. 2341–2356, 2018.

[Ch18b] Chatterjee, K.; Henzinger, M.; Loitzenbauer, V.; Oraee, S.; Toman, V.: Symbolic Algorithms for Graphs and Markov Decision Processes with Fairness Objectives. In: CAV. S. 178–197, 2018.

[Ch62] Church, A.: Logic, arithmetic, and automata. In: ICM. S. 23–35, 1962.

[DJL18] Daviaud, L.; Jurdzinski, M.; Lazic, R.: A pseudo-quasi-polynomial algorithm for mean-payoff parity games. In (Dawar, A.; Grädel, E., Hrsg.): LICS. ACM, S. 325–334, 2018, URL: https://doi.org/10.1145/3209108.3209162.

[JM20] Jurdzinski, M.; Morvan, R.: A Universal Attractor Decomposition Algorithm for Parity Games. CoRR abs/2001.04333/, 2020, arXiv: 2001.04333, URL: https://arxiv.org/abs/2001.04333.

[Ri53] Rice, H. G.: Classes of Recursively Enumerable Sets and Their Decision Problems. Transactions of the American Mathematical Society 74/2, S. 358–366, 1953, ISSN: 00029947, URL: http://www.jstor.org/stable/1990888.

[Sv22] Svozil, A.: Leveraging the Power of Graph Algorithms: Efficient Algorithms for Computer-Aided Verification, 2022, arXiv: 2202.02660.

[Tu37] Turing, A. M.: On computable numbers, with an application to the Entscheidungsproblem. Proceedings of the London mathematical society 2/1, S. 230–265, 1937.

[Va85] Vardi, M. Y.: Automatic Verification of Probabilistic Concurrent Finite-State Programs. In: FOCS. S. 327–338, 1985.

Alexander Svozil, geboren 1992, hat während dem Studium der Wirtschaftsinformatik seine Leidenschaft zur theoretischen Informatik entdeckt und absolvierte deswegen stattdessen das Bachelorstudium Software and Information Engineering auf der TU Wien weil dort die erweiterten Algorithmenvorlesungen im Lehrplan standen. Danach absolvierte er das Masterstudium Computational Intelligence ebenso auf der TU Wien. Von 2017 bis 2021 promovierte er begeistert an der Universität Wien in der Forschungsgruppe “Theorie und Anwendung von Algorithmen”, betreut von Prof. Monika Henzinger. Während dieser Zeit beschäftigte er sich mit fundamentalen Graphproblemen, Spielen auf Graphen, formaler Verifikation und Clustering Algorithmen. Seine Dissertation wurde von der Universität Wien für den österreichischen Staatspreis “Award of Excellence” nominiert. Seit 2021 arbeitet er bei Amazon als angewandter Wissenschaftler.

Inkrementalisierung Statischer Analysen in Datalog[1]

Tamás Szabó[2]

Abstract: Integrierte Entwicklungsumgebungen verwenden statische Analysen, um den Entwicklern bei der Bearbeitung ihrer Programme ein verwertbares Feedback zu geben. Im Gegenzug können die Entwickler ihren Code überarbeiten und potenzielle Laufzeitprobleme beseitigen, bevor der Code in Produktion geht. Die Entwicklung von Analysen für den Einsatz in IDEs ist ein komplexes Unterfangen, da die Analysen nach einer Programmänderung in Sekundenschnelle Ergebnisse liefern und gleichzeitig das Programmverhalten für alle möglichen Ausführungsarten präzise erfassen müssen. Diese beiden Anforderungen stehen im Widerspruch zueinander und stellen eine komplexe Herausforderung dar.

Die vorliegende Dissertation untersucht, wie sich Inkrementalität zur Beschleunigung statischer Analysen nutzen lässt. Als Reaktion auf eine Programmänderung berechnet eine inkrementelle Analyse nur die Ergebnisse neu, die von der Änderung betroffen sind, und verwendet den Rest der vorherigen Ergebnisse wieder. Die Dissertation beschreibt dazu den Entwurf und die Realisierung eines neuen Frameworks, das statische Analysen automatisch inkrementell ausführen kann. Die Dissertation zeigt, dass sich mit diesem Framework erhebliche Leistungsgewinne erzielen lassen und so selbst anspruchsvolle interprozedurale Analysen auf großen Softwaresystemen in wenigen Millisekunden inkrementell ausgeführt werden können.

1 Die Rolle der Statischen Analyse in der Softwareentwicklung

Statische Analysen sind eine Methode, um Rückschlüsse auf die Struktur oder das Verhalten eines Programms zu ziehen ohne es tatsächlich auszuführen. Dies steht im Gegensatz zu dynamischen Verifizierungstechniken wie Software-Tests oder Programm-Slicing, bei denen ein Programm ausgeführt werden muss. Angesichts der Tatsache, dass reale Softwaresysteme oft komplexe Abhängigkeiten aufweisen oder spezielle Hardware zur Ausführung benötigen (zum Beispiel GPUs oder Cloud-Umgebungen), hat die Fähigkeit, statische Schlüsse über das betreffende Programm zu ziehen, das Potenzial, Laufzeitfehler bereits zur Entwicklungszeit zu erkennen. Dies wiederum hilft, die Qualität der Software zu verbessern, zum Beispiel in Bezug auf Sicherheit oder Leistung, bevor die Software tatsächlich in Produktion geht. Dadurch können erhebliche Kosten für potenziell schädliche Auswirkungen von Softwarefehlern eingespart werden [Kr18].

Statische Analysen kommen in vielen Bereichen der modernen Softwareentwicklung vor. Integrierte Entwicklungsumgebungen (Integrated Development Environments, IDEs) verwenden häufig statische Analysen um den Entwicklern automatisches Feedback zu den betreffenden Programmen zu geben, indem sie Typüberprüfungen oder Datenflussanalysen durchführen. Sie tun dies, indem sie alle möglichen Eingaben eines Programms und alle

[1] Englischer Titel der Dissertation: “Incrementalizing Static Analyses in Datalog”

[2] Johannes Gutenberg-Universität Mainz, Institut für Informatik, Staudingerweg 9, 55128 Mainz, Deutschland
szabta89@gmail.com

möglichen Ausführungen berücksichtigen. Continuous Integration (CI)-Server verwenden statische Analysen, um Pull-Requests auf Verstöße gegen Kodierungsstandards zu kommentieren. Compiler verwenden statische Analysen, um Optimierungen wie die Konstantenpropagation und andere semantikerhaltende Code-Transformationen anzuwenden.

Ein konkretes Beispiel ist eine IDE, die den Programmierer vor unerreichbarem Code (dead code) in C Programmen warnt. Ein Teil eines Programms gilt als unerreichbar, wenn er unabhängig von der Eingabe bei keiner Ausführung erreicht werden kann. Ein bekannter Fehler, der mit unerreichbarem Code zusammenhängt, ist *goto fail*, von dem 2014 alle iPhone-Geräte betroffen waren [14]. Der Fehler lag in der Implementierung einer C-Funktion, die SSL-Schlüsselüberprüfung durchführt. Der relevante Ausschnitt der Funktion sieht wie folgt aus:

```
1  static OSStatus SSLVerifySignedServerKeyExchange(...) {
2    ...
3    if (...)
4      goto fail;
5      goto fail;
6    if (...)
7      goto fail;
8    err = sslRawVerify(ctx,
9      ctx->peerPubKey,
10     dataToSign,     /* plaintext */
11     dataToSignLen,  /* plaintext length */
12     signature,
13     signatureLen);
14   if(err) {
15     sslErrorLog(...);
16     goto fail;
17   }
18   fail:
19     SSLFreeBuffer(&signedHashes);
20     SSLFreeBuffer(&hashCtx);
21     return err;
22 }
```

Durch die Duplizierung von Zeile 4 haben die Entwickler versehentlich eine unbedingte `goto` Anweisung in Zeile 5 eingeführt, d.h. immer wenn die Programmausführung Zeile 5 erreicht, springt die Ausführung zum Label `fail` ohne Vorbedingung. Keiner der Codes in den Zeilen 6-17 ist erreichbar, unabhängig von der Programmeingabe. Dies ist ein Problem, weil die Zeilen 6-17 wichtige Überprüfungsschritte durchführen, insbesondere die Funktion `sslRawVerify`, aber der Funktionsaufruf ist Teil des nicht erreichbaren Codes. Ohne diese Überprüfungsschritte konnten Angreifer auf sensible Benutzerdaten zugreifen, selbst wenn die Kommunikation durch SSL/TLS geschützt war.

`https://opensource.apple.com/source/Security/Security-55471/libsecurity_ssl/lib/sslKeyExchange.c.auto.html`.

Die obige Analyse von unerreichbarem Code ist ein nützliches Hilfsmittel für Softwareentwickler, weil die Entwickler ihren Code auf der Grundlage des Feedbacks überarbeiten können um die Codequalität zu verbessern. Jedoch ist die Integration einer solchen Analyse in IDEs ein schwieriges Unterfangen. Einerseits haben IDEs *strenge Zeitbeschränkungen*; Analysen, die im Hintergrund laufen, haben höchstens ein paar Dutzend Millisekunden Zeit, um ihre Ergebnisse nach einer Codeänderung zu aktualisieren, andernfalls unterbrechen sie den Entwicklungsfluss, was schnell dazu führt, dass die Entwickler die Analysen nicht mehr nutzen [Jo13]. Andererseits erwarten die Entwickler, dass die Analysen so *präzise* wie möglich sind, d.h. dass sie alle möglichen Verhaltensweisen des Programms genau erfassen, und zwar auf eine *skalierbaren* Weise für reale Programmgrößen. Leider stehen diese Anforderungen im Widerspruch zueinander, und es ist eine komplexe Aufgabe, sie auszugleichen.

Das Ziel meiner Dissertation [Sz21] ist es, zu untersuchen, wie wir statische Analysen von realistischen Programmen so beschleunigen können, dass sie als Grundlage für kontinuierliches Feedback in IDEs dienen können. Konkret zielen wir darauf ab, die Landschaft der statischen Analysen zu verändern, indem wir Laufzeiten von unter einer Sekunde für Analysen ermöglichen, die ursprünglich nicht für den Einsatz in IDEs geeignet waren, weil ihre hochpräzise Natur nicht mit schnellen Aktualisierungszeiten vereinbar war oder weil realistische Programme allein aufgrund ihrer Größe eine Herausforderung darstellten.

In dieser Dissertation untersuchen wir, wie man *Inkrementierung* zur Beschleunigung statischer Analysen einsetzen kann. Die Hauptidee der Inkrementalisierung ist die effiziente Wiederverwendung von zuvor berechneten Analyseergebnissen. Abbildung 1 veranschaulicht, wie eine inkrementelle Analyse im Vergleich zu einer nicht-inkrementellen Analyse funktioniert. Angenommen, wir führen die Analyse von unerreichbarem Code für $\texttt{Programm}_1$ durch. Die Analyse erzeugt das anfängliche $\texttt{Ergebnis}_1$ in Form einer Menge, die aus den Labels der unerreichbaren Anweisungen besteht. Dann ändert ein Entwickler $\texttt{Programm}_1$, indem er Anweisungen zum Programm hinzufügt, entfernt oder austauscht um $\texttt{Programm}_2$ zu erhalten. Das Delta zwischen den beiden Programmen besteht aus der Menge der eingefügten und gelöschten Anweisungen, die wir mit $\Delta(\texttt{Programm}_1)$ bezeichnen. Ohne Inkrementalisierung müsste $\texttt{Programm}_2$ von Grund auf neu analysiert werden, um das neue Ergebnis $\texttt{Ergebnis}_2$ zu erhalten. Im Gegensatz dazu verwendet eine inkrementelle Analyse die alten Ergebnisse $\texttt{Ergebnis}_1$ wieder und analysiert nur die geänderten Codeteile $\Delta(\texttt{Programm}_1)$, um $\Delta(\texttt{Ergebnis}_1)$ zu erhalten, welches exakt der Differenz zwischen $\texttt{Ergebnis}_1$ und $\texttt{Ergebnis}_2$ entspricht. Diese Wiederverwendung ist genau das, was zu erheblichen Leistungsverbesserungen gegenüber der wiederholten Neuanalyse des gesamten Programms führen kann.

Wir haben uns für die Inkrementalisierung als Methode entschieden, da dies gut zu der Arbeit von Entwicklern in IDEs passt. Entwickler machen typischerweise eine Großzahl von *kleinen* Änderungen am Programm bis zum gewünschten Ergebnis. Die Hoffnung ist, dass kleine Programmänderungen inkrementell effizient verarbeitet werden können und auch nur geringfügige Auswirkungen auf das Ergebnis von Analysen haben werden. Während ein

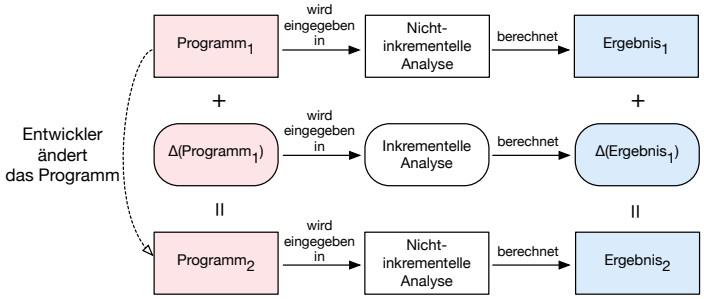


Abb. 1: Eine inkrementelle Analyse (abgerundete Ecken) berechnet das gleiche Ergebnis wie eine nicht-inkrementelle Analyse (Rechtecke), sie ist dabei jedoch um ein vielfaches schneller.

solcher Zusammenhang von kleinen Eingabeänderung zu kleinen Ausgabeänderung erfolgreich zum Beispiel in der algorithmischen Geometrie [Ka84], im Machine Learning [Su08], oder im Big Data Processing [Bh11] gezeigt werden konnte, ist es für statische Analysen unklar, ob sich hier auch erhebliche Leistungsverbesserungen erzielen lassen.

Auch technisch ist der Einsatz von Inkrementalisierung zur Beschleunigung statischer Analysen eine große Herausforderung. Wir haben oben eine Reihe von Beispielen beschrieben, bei denen die Inkrementalisierung erfolgreich funktionierte, aber es muss darauf hingewiesen werden, dass inkrementelle Analysen oft Einzellösungen mit spezialisierten Algorithmen sind. Häufig ist es nicht offensichtlich, wie man eine gute inkrementelle Leistung *im Allgemeinen*, also für beliebige statische Analysen, erreichen kann. Darüber hinaus gibt es eine Reihe von statische Analyse-spezifischen technischen Herausforderungen, die eine effiziente Inkrementalisierung erschweren. Wir erörtern diese Herausforderungen im Folgenden.

2 Herausforderungen bei der Inkrementalisierung Statischer Analysen

Die Inkrementalisierung statischer Analysen muss zwei Eigenschaften erfüllen: Sie müssen effizient und korrekt sein. Als Reaktion auf eine Programmänderung muss eine *korrekte* inkrementelle Analyse die vorherigen Ergebnisse so aktualisieren, dass das aktualisierte Ergebnis dem Ergebnis einer konventionell nicht-inkrementellen Analyse exakt entspricht. Mit Blick zurück auf Abbildung 1 erfordert die Korrektheit, dass $\texttt{Ergebnis}_2$ dasselbe ist wie $\texttt{Ergebnis}_1 + \Delta(\texttt{Ergebnis}_1)$. Dabei muss die amortisierte Laufzeit für die inkrementelle Ausführung proportional zu der Größe der Programmänderung $\Delta(\texttt{Programm}_1)$ sein, damit kleine Änderungen auch tatsächlich schnelle Ergebnisse nach sich ziehen. Wir nennen solche inkrementelle Analysen *effizient inkrementell*. Für praktisch relevante statische Analysen, die das Programmverhalten interprozedural erfassen, ist jedoch unklar, ob eine effizient inkrementelle Definition überhaupt existiert. Die vorliegende Dissertation beantwortet diese Frage positiv und geht noch einen Schritt weiter, indem sie ein automatisiertes Verfahren

vorstellt, mit dem korrekte, effizient inkrementelle statische Analysen aus deklarativen Beschreibungen abgeleitet werden können. Dabei mussten folgende Herausforderungen überwunden werden:

- *Inkrementelle Verarbeitung von strukturierten Daten*: Typischerweise arbeiten statische Analysen mit einer strukturierten Darstellung des betreffenden Programms, wie z.B. dem abstrakten Syntaxbaum oder dem Kontrollflussgraphen.Für die Inkrementalisierung ist es jedoch entscheidend, wie die interne Darstellung dieser Datenstrukturen aussieht. Wenn wir beispielsweise Knoten mit Indizes oder Labels modellieren, die von einer Art globaler Ordnung abhängen (z. B. absolute Position im Programm), dann kann eine Programmänderung aufgrund einer Indexverschiebung unnötig viele Knoten betreffen.
- *Komplexe rekursive Abhängigkeiten*: Eine inkrementelle statische Analyse erfordert die Verfolgung von Abhängigkeiten, um zu erkennen, welcher Teil des Analyseergebnisses nach einer Programmänderung neu berechnet werden muss. Präzise Analysen haben jedoch viele rekursive Komponenten, die miteinander verwoben sind, was die Abhängigkeitsverfolgung erheblich erschwert. Zum Beispiel erfordert die Analyse von unerreichbarem Code die Konstruktion eines Kontrollflussgraphen, jedoch auch die Auflösung von Funktionszeigern durch eine Points-to-Analyse. Dabei hängen Kontrollfluss- und Points-to-Analyse von einander wechselseitig ab und verstärken sich gegenseitig. Wenn sich ein betroffenes Programm ändert, müssen wir solch rekursive Abhängigkeiten sorgfältig behandeln, um das Analyseergebnis korrekt zu aktualisieren.
- *Vollständige Verbände und rekursive Aggregation*: Statische Analysen verwenden routinemäßig benutzerdefinierte Verbände für die Darstellung des abstrakten Bereichs [NNH15]. Verbände sind teilweise geordnete Mengen mit Aggregationsoperatoren zur Berechnung der kleinsten oberen Grenze oder der größten Untergrenze von zwei beliebigen abstrakten Werten. Diese Aggregationsoperatoren spielen eine wichtige Rolle bei der Analyse von Programmen, da hiermit Informationen auf verschiedenen Kontrollflusspfaden kombiniert und approximiert werden. Auf Grund des zyklischen Kontrollflusses von Programmen müssen Aggregationsoperatoren auch rekursiv anwendbar sein bis sich das Analyseergebnis stabilisiert. Dies erschwert die effiziente Wiederverwendung früherer Ergebnisse in einer inkrementellen Umgebung deutlich, da eine Änderung im betreffenden Programm ein vollständiges Zurückrollen der vorherigen rekursiv berechneten Ergebnisse erfordern kann.
- *Interprozeduralität*: Die Analyse über Funktionsaufrufe hinweg eröffnet ein ganz neues Feld an Herausforderungen für eine effiziente Inkrementierung, da selbst eine kleine Programmänderung große Auswirkungen auf das Gesamtergebnis der Analyse haben kann. So kann es beispielsweise leicht passieren, dass ein Programmierfehler in einer häufig verwendeten Bibliotheksfunktion dazu führt, dass Code in einer großen Zahl von Aufrufstellen nicht mehr erreichbar ist. In diesem Fall wäre die inkrementelle Laufzeit nicht mehr proportional zur Größe der Programmänderung, die Berechnung also nicht effizient inkrementell. Kann man dennoch erhebliche Verbesserungen der Laufzeit durch die Inkrementalisierung von Analysen erzielen?

Im Folgenden nennen wir Analysen *anspruchsvoll*, wenn sie alle oben genannten Features verwenden: strukturierte Daten, rekursive Abhängigkeiten, rekursive Aggregation, interprozedural. Darauf basierend vertritt die vorliegende Dissertation folgende These:

> Die Laufzeit anspruchsvoller statischer Analysen kann durch das Inkrementalisierung erheblich und automatisiert verbessert werden.

3 Bewältigung der Herausforderungen in der Dissertation

Wir zeigen, dass Inkrementalisierung die Leistung anspruchsvoller statischer Analysen erheblich verbessern kann, indem wir in dieser Dissertation ein inkrementelles statisches Analyse-Framework entwickeln und implementieren. Wir nennen unser Framework IncA, eine Abkürzung für *INC*remental *A*nalysis. IncA bietet eine deklarative Programmiersprache für Analyseentwickler und inkrementiert automatisch deren Analysen. Das heißt, IncA befreit Analyseentwickler von der technischen Komplexität der Inkrementalisierung. IncA macht keine Annahmen oder Einschränkungen in Bezug auf die Präzision der Analysen; den richtigen Kompromiss zwischen diesen zu finden, bleibt die Aufgabe des Analyseentwicklers.

Technisch gesehen verfügt IncA über einen optimierenden Compiler, der die deklarative Spezifikation einer Analyse nach Datalog übersetzt. Datalog ist eine logische Programmiersprache, jedoch wurden bislang ausschließlich nicht-inkrementelle statische Analysen in Datalog betrachtet. Meine Dissertation zeigt, wie Datalog verwendet und erweitert werden kann, um anspruchsvolle inkrementelle statische Analysen zu unterstützen.

Im Folgenden rekapitulieren wir, wie IncA die in Abschnitt 2 beschriebenen Herausforderungen löst:

- *Inkrementelle Verarbeitung von strukturierten Daten*: Ein Datalog-Programm verarbeitet flache Daten, die in Relationen, d.h. Mengen von Tupeln, organisiert sind. Wir haben eine Kodierung von strukturierten Daten in Datalog entwickelt, die eine inkrementelle Verarbeitung von Syntaxbäumen und deren Modifikation ermöglicht. Insbesondere modelliert IncA Syntaxbäume als Knoten mit Kanten, da dies eine maximale Entkopplung von Teilbäumen ermöglicht. So ist es zum Beispiel möglich, den Rückgabetyp einer Methode (eine Kante) zu ändern, ohne den Methodenkörper (eine andere Kante) als geändert zu markieren. Da das Schreiben komplexer Analysen gegen ein solches entkoppeltes Datenformat mühsam und fehleranfällig wäre, ist es von entscheidender Bedeutung, dass die Architektur von IncA eine deklarative, benutzerorientierte Sprache enthält, die diese technischen Einzelheiten abstrahiert. So erlaubt IncA den Entwicklern von Analysen die Verwendung von konventionellem Pattern Matching, um strukturierte Daten zu verarbeiten, und kompiliert dies zu komplexen Abfragen gegen Relationen, die die Knoten und Kanten des Baums speichern.

- *Komplexe rekursive Abhängigkeiten*: Bei Analysen in der realen Welt wird häufig Rekursion verwendet, da reale Programme typischerweise einen zyklischen Kontrollfluss haben. Auch wenn es bestehende Algorithmen zur Inkrementierung von rekursivem Datalog gibt, eignen sie sich nicht unmittelbar um statische Analysen, die in Datalog kodiert sind, effizient zu inkrementieren. In der Tat zeigen wir zum ersten Mal, dass es möglich ist, komplexe Analysen systematisch und automatisch zu inkrementieren. Technisch gesehen verwendet IncA ein Berechnungsnetzwerk, das auf Programmänderungen mit feinkörniger Änderungspropagation reagieren kann und dabei möglichst viele der Analyseergebnisse wiederverwendet.
- *Vollständige Verbände und rekursive Aggregation*: Bestehende inkrementelle Datalog-Solver unterstützen Rekursion, setzen aber voraus, dass alle Daten endlich sind. Dies ist unvereinbar mit statischen Analysen, bei denen Verbandswerte berechnet und aggregiert werden, die sich über unendliche Bereiche erstrecken. Um realistische statische Analysen zu unterstützen, haben wir einen neuartigen inkrementellen Datalog-Solver entwickelt, der rekursive Aggregationen über unendliche Verbände unterstützt. IncA nutzt die algebraischen Eigenschaften von Verbänden, um monoton wachsende Aggregationen zu erkennen und verarbeitet sie als In-Place-Updates. Dies ist eine grundlegende Erweiterung von Datalog und ermöglicht es Datalog zum ersten Mal, unendliche benutzerdefinierte Daten inkrementell zu aggregieren. Auf dieser Grundlage können IncA-Benutzer eigene Verbände in ihren rekursiven Analysedefinitionen einsetzen und IncA inkrementiert diese automatisch.
- *Interprozeduralität*: Bevor versucht wird, die effiziente inkrementelle Wartung von interprozeduralen Analysen technisch zu lösen, tritt die Dissertation eigentlich einen Schritt zurück und beantwortet die Frage, ob dies überhaupt möglich ist. Erinnern wir uns daran, dass die Inkrementierung zum Scheitern verurteilt ist, wenn die Wiederverwendung von zuvor berechneten Analyseergebnissen nicht möglich ist, weil eine Programmänderung die Ausgabe erheblich verändern würde. Wir haben ein groß angelegtes Experiment durchgeführt, in dem wir untersucht haben, wie eine Reihe von interprozeduralen Analysen auf Programmänderungen reagieren, d. h. wie viele der von ihr berechneten Analyseergebnissen verworfen werden müssen. Es stellte sich heraus, dass nach kleinen Programmänderungen in der überwiegenden Mehrheit der Fälle auch nur kleine Änderungen in den Analyseergebnissen zu beobachten sind, was bei interprozeduralen Analysen doch überrascht. Dies ist eine wichtige Voraussetzung für eine mögliche Lösung, da es nun am Datalog-Solver liegt, dieses Verhalten tatsächlich auszunutzen. Wir fanden heraus, dass der Einsatz einer Technik namens Differential Dataflow (DDF) [Mc13] effizient bei der inkrementellen Aufrechterhaltung von interprozeduralen Analysen eingesetzt werden kann, allerdings unterstützt DDF wiederum keine Verbände. IncA hebt DDF auf die nächste Stufe, indem es schnelle Aktualisierungen auch für verbandbasierte Berechnungen ermöglicht.

4 Hauptbeiträge der Dissertation

Theoretische Ergebnisse Diese Dissertation bringt den Stand der Technik in der statischen Analyse und auch in Datalog erheblich voran. Was die statische Analyse betrifft, so ist die Tatsache, dass IncA rekursive Datalog-Programme mit verbandbasierten Aggregationen unterstützt, eine bedeutende Verbesserung der Ausdruckskraft. Datalog ist bereits eine beliebte Wahl unter den Praktikern statischer Analysen, aber die verbesserte Ausdruckskraft ermöglicht eine ganze Reihe neuer Anwendungen, die ohne spezielle Unterstützung für benutzerdefinierte Verbände entweder unpraktikabel oder ineffizient waren. Zweitens werden wir, wenn es um Datalog geht, aus der statischen Analyse herauszoomen, da dies nur ein Anwendungsbereich von IncA ist. Ganz allgemein bringt IncA den Stand der Technik in Datalog voran, indem es auf weit verbreiteten Solver-Algorithmen aufbaut und diese erweitert. Vor IncA konnten Datalog-Solver nur primitive Werte (z.B. Strings oder Zahlen) zur Laufzeit generieren, aber die Unterstützung von verbandbasierten Aggregationen bringt hier einen deutlichen Fortschritt. Darüber hinaus beweisen wir formal, dass unsere neuen inkrementellen Algorithmen genau die gleichen Ergebnisse berechnen, die ihre nicht-inkrementellen Gegenstücke auf der gleichen Eingabe von Grund auf berechnen würden. Dies zeigt, dass wir die Korrektheitsanforderung erfüllen, die wir in Abschnitt 2 diskutiert haben.

Praktische Ergebnisse IncA ist das erste statische Analyse-Framework, das Datalog-Programme mit verbandbasierter rekursiver Aggregation inkrementiert. Tatsächlich ist das Design von IncA unabhängig von der jeweiligen Sprache oder Analyse, und abgesehen von einer kleinen IDE-spezifischen Schnittstelle ist auch die Solver-Engine IDE-unabhängig wiederverwendbar. Dies ist ein zentrales Designprinzip von IncA, dass es uns ermöglichte, sowohl parser-basierte (textuelle) als auch projektionale IDEs zu unterstützen. Der Entwurf und die Implementierung von IncA wurden während der gesamten Forschungsarbeit durch Laufzeitmessungen begleitet und dafür optimiert. Wir haben eine Vielzahl von statischen Analysen als Benchmarks implementiert, die von einfachen Wohlgeformtheitsregeln bis hin zu vollständigen interprozeduralen Points-to-Analysen für Java reichen. Wir zeigen empirisch, dass IncA selbst bei anspruchsvollen Analysen Aktualisierungszeiten von unter einer Sekunde nach Programmänderungen liefern kann, während es akzeptable Initialisierungszeiten und einen akzeptablen Speicherverbrauch aufweist. Jeder Aspekt der Forschungsarbeit führte zu Software-Prototypen, und diese sind alle als Open-Source unter `https://github.com/szabta89/IncA` verfügbar.

Ausblick Die Kernkomponenten von IncA wurden in Zusammenarbeit mit einem Forschungsingenieur eines industriellen Partners entwickelt, und unsere algorithmischen Beiträge sind verfügbar im offiziellen Viatra Queries Eclipse Projekt. Wir glauben, dass IncA als Framework ausgereift genug ist, um für reale Anwendungen in IDEs verwendet zu werden. Gegen Ende der Doktorarbeit des Autors wurde besonderer Wert darauf gelegt,

IncQuery Labs, `https://incquerylabs.com/`
`https://www.eclipse.org/viatra/`

die Forschungsergebnisse besser zugänglich zu machen, und dieser Wunsch trieb die Untersuchung der Integration von IncA in textuelle IDEs im Gegensatz zu projektionalen IDEs an, da textuelle IDEs als Mainstream angesehen werden können. Dies ist keine einfache Aufgabe, aber die Dissertation zeigt, wie man diese Herausforderung lösen kann. Die Grundlagen von IncA ermöglichten auch einem anderen Doktoranden an der JGU Mainz eine weiterführende Forschungsarbeit, die bereits zu weiteren Veröffentlichungen bei OOPSLA'20 [PES20] und ECOOP'22 [PE22] führte.

Forschungsergebnisse Die Dissertation resultierte in 4 Veröffentlichungen auf internationalen Konferenzen von höchstem Rang: ASE'16 [SEV16], OOPSLA'18 [Sz18a], PLDI'21 [SEB21], PLDI'21 [ESP21]. Weitere Beiträge der Dissertation wurden bei einem internationalen Workshop FTfJP'18 [Sz18b] sowie als Gastbeitrag der Fachkonferenz für logische Programmierung ICLP'20 [Sz20] vorgestellt.

Literatur

[14] CVE-2014-1266, `http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-1266`, 2014.

[Bh11] Bhatotia, P.; Wieder, A.; Rodrigues, R.; Acar, U. A.; Pasquin, R.: Incoop: MapReduce for Incremental Computations. In: Symposium on Cloud Computing (SoCC). ACM, 2011.

[ESP21] Erdweg, S.; Szabó, T.; Pacak, A.: Concise, Type-Safe, and Efficient Structural Diffing. In: Programming Language Design and Implementation (PLDI). ACM, S. 406–419, 2021.

[Gr13] Green, T. J.; Huang, S. S.; Loo, B. T.; Zhou, W.: Datalog and Recursive Query Processing. Foundations and Trends in Databases 5/2, S. 105–195, 2013.

[Jo13] Johnson, B.; Song, Y.; Murphy-Hill, E.; Bowdidge, R.: Why Don't Software Developers Use Static Analysis Tools to Find Bugs? In: International Conference on Software Engineering (ICSE). IEEE Press, S. 672–681, 2013.

[Ka84] Kallay, M.: The complexity of incremental convex hull algorithms in Rd. Information Processing Letters 19/4, S. 197, 1984.

[Kr18] Krasner, H.: The Cost of Poor Quality Software in the US: A 2018 Report, `https://perma.cc/TQN7-A68B`, accessed on January 4, 2020, 2018.

[Mc13] McSherry, F.; Murray, D. G.; Isaacs, R.; Isard, M.: Differential Dataflow. In: Conference on Innovative Data Systems Research (CIDR). 2013.

[NNH15] Nielson, F.; Nielson, H. R.; Hankin, C.: Principles of program analysis. Springer, 2015.

[PE22] Pacak, A.; Erdweg, S.: Functional Programming with Datalog. In: European Conference on Object-Oriented Programming (ECOOP). 2022.

[PES20] Pacak, A.; Erdweg, S.; Szabó, T.: A Systematic Approach to Deriving Incremental Type Checkers. Proceedings of the ACM on Programming Languages 4/OOPSLA, 2020.

[SEB21] Szabó, T.; Erdweg, S.; Bergmann, G.: Incremental Whole-Program Analysis in Datalog with Lattices. In: Programming Language Design and Implementation (PLDI). ACM, S. 1–15, 2021.

[SEV16] Szabó, T.; Erdweg, S.; Voelter, M.: IncA: A DSL for the Definition of Incremental Program Analyses. In: Automated Software Engineering. ACM, S. 320–331, 2016.

[Su08] Sumer, O.; Acar, U.; Ihler, A. T.; Mettu, R. R.: Efficient Bayesian Inference for Dynamically Changing Graphs. In (Platt, J. C.; Koller, D.; Singer, Y.; Roweis, S. T., Hrsg.): Advances in Neural Information Processing Systems. Curran Associates, Inc., S. 1441–1448, 2008.

[Sz18a] Szabó, T.; Bergmann, G.; Erdweg, S.; Voelter, M.: Incrementalizing Lattice-based Program Analyses in Datalog. Proceedings of the ACM on Programming Languages 2/OOPSLA, 139:1–139:29, 2018.

[Sz18b] Szabó, T.; Kuci, E.; Bijman, M.; Mezini, M.; Erdweg, S.: Incremental Overload Resolution in Object-oriented Programming Languages. In: Companion for ISSTA/ECOOP 2018 Workshops. ACM, S. 27–33, 2018.

[Sz20] Szabó, T.; Bergmann, G.; Erdweg, S.; Voelter, M.: Datalog with Recursive Aggregation for Incremental Program Analyses. In: International Conference on Logic Programming (ICLP)–Sister Conferences and Journal Presentation Track. 2020.

[Sz21] Szabó, T.: Incrementalizing Static Analyses in Datalog, Diss., Universitätsbibliothek der Johannes Gutenberg-Universität Mainz, 2021.

Tamás Szabó, geboren am 26.03.1989 in Debrecen, Ungarn, hat von 2007 bis 2013 Informatik an der TU Budapest studiert. Im Jahr 2015 begann er sein Promotionsstudium an der TU Darmstadt unter der Betreuung von Prof. Sebastian Erdweg. Während seiner akademischen Laufbahn begleitete er Prof. Erdweg und war Doktorand an der TU Delft in den Niederlanden und schloss seine Promotion im Januar 2021 an der JGU Mainz ab. Parallel zu seinem Promotionsstudium arbeitete er als Softwareentwickler bei der itemis AG in Vollzeit und anschließend bei Workday. Seine Forschungsinteressen konzentrieren sich auf den Entwurf und die Implementierung von Programmiersprachen, statische Programmanalysen und inkrementelle Berechnungen.

Kryptographische Schlösser für Skriptlose Kryptowährungszahlungen[1]

Sri AravindaKrishnan Thyagarajan[2]

Abstract: Blockchain-basierte Kryptowährungen haben in den letzten zehn Jahren als eine Form des vertrauenslosen und dezentralen Zahlungsmechanismus' an Bedeutung gewonnen. Für diese Währungen wurden Zahlungsprotokolle entwickelt, die entweder darauf abzielen, die Leistung der Währungen zu verbessern, oder sie als Teil einer größeren Anwendung zu nutzen. Mehrere dieser Protokolle leiden jedoch unter Datenschutz-, Kompatibilitäts- oder Kostenproblemen, wenn sie für verschiedene Währungen verwendet werden. In dieser Dissertation entwickeln wir neue Zahlungsprotokolle sowohl für Anwendungen als auch für Leistungsverbesserungen, die eine bessere Privatsphäre für die Benutzer garantieren, mit allen Währungen kompatibel sind und den Benutzern geringere Kosten bieten. Mehrere der in dieser Dissertation entwickelten Protokolle können in vielen der bestehenden Währungen eingesetzt werden und sind auch für Währungen geeignet, die in der Zukunft entwickelt werden könnten.

1 Einführung

Bitcoin [Na08] wurde 2008 erfunden und war das erste dezentralisierte Währungssystem, das verschiedene kryptografische Mechanismen und ein neuartiges verteiltes Werkzeug zur Konsensbildung namens *Blockchain* verwendet. Zahlungen zwischen Nutzern können so getätigt werden, dass keine zentrale Behörde wie eine Bank erforderlich ist, um diese Zahlungen zu genehmigen. Stattdessen werden diese Zahlungen auf öffentlich überprüfbare Weise über das dezentralisierte Zahlungsregister, kurz Blockchain genannt, validiert. Darüber hinaus gewährleistet die verwendete Konsens-Technologie die Unveränderlichkeit der Zahlungen, d. h. diese Zahlungen selbst können nicht manipuliert werden; insbesondere kann ein Nutzer nicht zweimal dieselbe Münze oder die Münze eines anderen ausgeben.

Litecoin, Ethereum [Wo14], Zcash [Sa14] und Monero [La19] sind einige der beliebtesten Kryptowährungen, die eine ähnliche Philosophie verfolgen. Jede dieser Währungen bietet unterschiedliche Funktionalitäten oder Sicherheitsgarantien. Ethereum ermöglicht es den Nutzern zum Beispiel, komplexe bedingte Zahlungen untereinander zu tätigen, die als *Smart Contracts* bezeichnet werden. Zcash und Monero ermöglichen es Nutzern, Zahlungen unter Wahrung der Privatsphäre vorzunehmen, bei denen andere Nutzer weder den Absender noch den Empfänger oder den Betrag einer Zahlung ermitteln können. All dies wird durch die Fähigkeit der Blockchain erleichtert, kryptografische Überprüfungsverfahren zu unterstützen, die selbst dezentral und öffentlich überprüfbar sind.

Mehrere Anwendungen basieren auf diesen Kryptowährungen, um deren Garantien auszunutzen. Zu diesen Anwendungen gehören ein nicht manipulierbarer Zeitstempeldienst,

[1] English Title of the dissertation: Cryptographic Locks for Scriptless Cryptocurrency Payments

[2] Carnegie Mellon University, t.srikrishnan@gmail.com

Fairness beim Austausch zwischen oder bei Berechnungen mit mehreren Parteien, vertrauenslose Auktionen, vertrauenslose *Bug-bounty* Programme und viele andere. Die derzeitigen Entwicklungen sprechen auch für eine weit verbreitete Übernahme der Technologie auch durch etablierte Akteure wie Accenture, Goldman Sachs, Google, Facebook usw. Diese Anwendungen beruhen im Wesentlichen auf der öffentlichen Überprüfbarkeit der Transaktionen in der Blockchain. Jeder Nutzer des Systems kann überprüfen, ob die Datensätze miteinander übereinstimmen. Solche Überprüfungen werden mit Hilfe von *Skripten* durchgeführt, d. h. mit Code in einer Sprache, die jedem Nutzer des Systems bekannt ist und die er auf seinem lokalen Rechner ausführen kann.

Genauer hat eine Transaktion zwei wichtige Felder: die Absenderinformation und die Empfängerinformation. Im häufigsten Fall der heutigen Kryptowährungen geben die Absenderinformationen die Adresse des Absenders (öffentlicher Schlüssel) an, und das Skript ist der Verifizierungscode des Algorithmus zur Überprüfung der digitalen Signatur, der zur Authentifizierung von Transaktionen verwendet wird. Die Absenderinformation enthält zusätzlich die digitale Signatur selbst, wobei die Signatur in Bezug auf den öffentlichen Schlüssel des Absenders, die Transaktion selbst und den Verifizierungscode gültig ist. Die Empfängerinformation enthält die Empfängeradresse (öffentlicher Schlüssel) und den Wert der an diese Adresse gesendeten Münzen. In den Empfängerinformationen wird auch eine Bedingung in Form eines Skripts angegeben, unter der die Münzen von dieser Empfängeradresse weiter ausgegeben werden können. Bei einer typischen eins-zu-eins-Übertragung zwischen zwei Nutzern wäre die Bedingung der Verifizierungsalgorithmus des digitalen Signatursystems. Um die Münzen von der Empfängeradresse auszugeben, muss der Empfänger daher eine weitere Transaktion mit dieser Adresse als Absenderadresse durchführen und eine entsprechende digitale Signatur hinzufügen.

Zu den anderen beliebten Skripten gehören t-`aus`-n-`Multisig`-Adress-Skripte, die t Signaturen aus n Signaturen erfordern, `timelock`-Skripte, die eine Zahlung erst ab einem bestimmten Zeitpunkt in der Zukunft gültig werden lassen, und *smart contracts*, die komplexe bedingte Zahlungen ermöglichen und viele weitere. Diese Skripte werden in allen auf Kryptowährungen basierenden Anwendungen und auch in Zahlungsprotokollen, die die Leistung und Sicherheit der zugrunde liegenden Währung selbst verbessern sollen, ausführlich genutzt.

1.1 Probleme bei der Verwendung von Skripten in Zahlungsprotokollen

Die Verwendung solcher Skripte in einem dezentralisierten öffentlichen Zahlungssystem bringt jedoch einige Probleme mit sich, die im Folgenden aufgeführt sind:

1. *Datenschutz oder Fungibilität der Zahlungen.* Zahlungsprotokolle, die Skripte verwenden, geben öffentlich Informationen über das Protokoll und seine Teilnehmer an andere Nutzer der Blockchain weiter. Wenn ein Protokoll außerdem Transaktionen mit speziellen Skripten verwendet, erhalten die mit diesen Transaktionen verbundenen Münzen einen *Pseudowert*. Dies ist schädlich für ein Währungssystem, weil es die *Fungibilität* der Münzen beeinträchtigt, da diese Münzen im Vergleich

zu anderen Münzen im System, die nur an Transaktionen mit regulären Skripten wie der Unterschriftenprüfung beteiligt sind, gekennzeichnet sind. Solche gekennzeichneten Münzen könnten beispielsweise als Zahlungsmittel abgelehnt werden, es könnten höhere Gebühren erhoben werden, oder sie könnten von den Nutzern des Währungssystems selektiv bevorzugt werden.

2. *Kompatibilität oder Interoperabilität.* Wie bereits angedeutet haben verschiedene Währungen unterschiedliche Designziele und bieten daher unterschiedliche durch Skripte unterstützte Zahlungen. Das Skript *Hash TimeLock Contracts (HTLC)* wird zum Beispiel in Bitcoin unterstützt, aber nicht in Monero oder Zcash, während *smart contracts* in Ethereum, aber nicht in Bitcoin untersützt werden. Ein konkretes Beispiel ist das Lightning Network, das *Payment Channel Networks (PCN)* als Skalierbarkeitslösung verwendet, die derzeit in Bitcoin eingesetzt wird. Es nutzt das spezielle (HTLC)-Skript, das in Bitcoin verfügbar ist. Die Lösung ist jedoch nicht mit Währungen wie Monero, Zcash, Mimblewimble usw. kompatibel.

3. *Hohe Kosten.* Die dezentrale Validierung von Zahlungen hat den Preis, dass jeder Nutzer im System jede Zahlung validieren muss. In letzter Zeit gab es Bemühungen, diesen Overhead zu amortisieren, indem nur eine ausgewählte Gruppe von Nutzern jede Zahlung validieren darf. Bei den derzeit genutzten Kryptowährungsnetzwerken muss jedoch jede Zahlung von Tausenden von Nutzern auf der ganzen Welt validiert werden. Weiterhin müssen zur Validierung einer Zahlung die Skripte ausgeführt werden, was je nach Komplexität recht ineffizient sein kann. Eine einfache Unterschriftenprüfung für ein Standard-Signaturverfahren ist viel billiger als ein Skript wie Mutlisig oder smart contracts. Diese offensichtliche Ineffizienz führt dazu, dass die Nutzer, die die Transaktionen erstellen, hohe Gebühren für die Validierung auf der Blockchain zahlen müssen. Außerdem führen größere Skripte zu einer schlechten Skalierbarkeit des gesamten Systems, was wiederum zu höheren Gebühren für jeden Nutzer im System führt.

Deshalb stellen wir in dieser Dissertation die Frage,

> *Können wir Zahlungsprotokolle entwerfen, die eine bessere Fungibilität, bessere Interoperabilität und geringere Kosten aufweisen?*

Bei der Beantwortung dieser Frage müssen wir mehrere Herausforderungen berücksichtigen:

1. Wir möchten den Datenschutz und die Sicherheit der Mittel für diese Protokolle verbessern oder zumindest auf den neuesten Stand bringen, und
2. wir möchten eine praktische Effizienz erreichen, die mit der ihrer skriptbasierten Gegenstücke vergleichbar ist.

Wir bemühen uns, diese Herausforderungen zu bewältigen, indem wir kryptografische Techniken einsetzen, die die erforderlichen Sicherheitsgarantien bieten und gleichzeitig sicherstellen, dass diese Techniken auch den Effizienzanforderungen gerecht werden.

2 Unsere Ergebnisse

Die Zahlungsprotokolle, die in dieser Arbeit von Interesse sind, werden derzeit unter Verwendung spezieller Skripte eingesetzt und dienen entweder der Verbesserung: (1) Skalierbarkeit der Währung und damit Verbesserung des Transaktionsdurchsatzes und (2) währungsübergreifende Zahlungen. Insbesondere konzentrieren wir uns auf Protokolle für *payment channels* und *payment channel networks*, die Lösungen sind, die skalierbare währungsübergreifende Zahlungen in Blockchain-basierten Kryptowährungen und *atomic swaps* von Münzen zwischen beliebigen Blockchain-basierten Kryptowährungen ermöglichen. Unser Ziel ist es, solche Protokolle zu entwerfen, die minimale Anforderungen an spezielle Skripte der zugrunde liegenden Blockchain haben.

2.1 Skalierbarer währungsübergreifender Zahlungsverkehr

Nehmen wir an, dass Benutzerin Alice Zahlungen an Bob leisten möchte. In einer normalen Umgebung postet Alice eine Transaktion auf der Blockchain. Wenn Alice jedoch mehrere Zahlungen über einen bestimmten Zeitraum hinweg leisten muss, ist das Posten einer Transaktion für jede Zahlung auf der Blockchain ziemlich teuer und führt zu großen Verzögerungen, da die Transaktionsverarbeitungsleistung der Blockchain begrenzt ist. Es ist bekannt, dass Bitcoin 7 Transaktionen pro Minute verarbeiten kann, während Visa-Zahlungen als traditionelles Zahlungssystem mehrere tausend Zahlungen pro Minute verarbeiten kann.

Zahlungskanäle sind eine Skalierbarkeitslösung, die für Kryptowährungen entwickelt wurde, um das oben genannte Problem zu lösen. Hier richten Alice und Bob zunächst eine gemeinsame Adresse ein, die *payment channel* genannt wird. Alice sendet einige Münzen an diese gemeinsame Adresse in Form einer Transaktion auf der Blockchain, die als *channel opening* Transaktion bezeichnet wird. Um Zahlungen vorzunehmen, sendet Alice von der gemeinsamen Adresse aus zahlende Transaktionen an Bob, aber keine dieser Transaktionen wird von Bob auf der Blockchain veröffentlicht. Diese Zahlungen werden als *off-chain payments* bezeichnet. Sobald Bob die Zahlungen abschließen möchte, wählt er eine Zahlungstransaktion aus, signiert sie zusätzlich zu Alice und postet sie auf der Blockchain, wodurch der Zahlungskanal geschlossen wird. Dies führt dazu, dass nur zwei Transaktionen in der Blockchain veröffentlicht werden, während möglicherweise mehrere Zahlungen von Alice an Bob getätigt wurden.

Payment Channel Networks (PCN) sind eine Verallgemeinerung der obigen Technik, bei der Alice statt eines direkten Zahlungskanals mit Bob einen mit einem anderen Nutzer Carol hat und Carol einen Zahlungskanal mit Bob. Die Zahlungen von Alice werden über Carol als Vermittler durch die beiden Zahlungskanäle geleitet. Diese Technologie wird bereits in Bitcoin in Form des Lightning Network (LN) [PD16] eingesetzt, das einen enormen Anstieg des Transaktionsdurchsatzes aufweist. Die aktuellen Vorschläge für PCN basieren entweder auf dem speziellen HTLC-Skript [PD16] oder erfordern ein Skript zur Überprüfung der Unterschrift, allerdings nur für eine bestimmte Klasse von Unterschriftsverfahren [Ma19]. Beide Varianten leiden unter mangelnder Interoperabilität, d. h., die

Nutzer können sich nur gegenseitig bezahlen, wenn sie bestimmte Währungssysteme verwenden. Der HTLC-Ansatz leidet zusätzlich unter der Fungibilität und den hohen On-Chain-Kosten.

In dieser Arbeit [Th22] entwerfen wir den ersten skalierbaren, währungsübergreifenden Payment Channel, der es Nutzern in jedem Währungssystem ermöglicht, einander auf vertrauenslose Weise zu bezahlen, vorausgesetzt, die Währungen unterstützen das minimale Signaturverifikationsskript und das `locktime`-Skript. Zu diesem Zweck konzentrieren wir uns auf Off-Chain-Zahlungsprotokolle, nämlich *payment channel (PC)* und seine Verallgemeinerung *payment channel network (PCN)* [PD16]. Wir entwerfen das erste generische Protokoll, das Zahlungen zwischen *beliebigen* Währungen mit *nur* einer Signaturprüfung der Währung für ein *beliebiges* Signaturschema abwickelt. Wir entwerfen auch hocheffiziente Protokolle für ausgewählte Signaturschemata von Interesse, für welche bisher kein PCN-Protokoll bekannt war.

Ein PCN-Protokoll ermöglicht zwei Nutzern, die unterschiedliche Währungen verwenden, eine Zahlung vorzunehmen, indem sie eine Reihe von Zwischenhändlern einsetzen, die diese Zahlung erleichtern. Die zwischengeschalteten Stellen können selbst unterschiedliche Währungen verwenden. Frühere Ansätze beruhen auf der Einrichtung von bedingten Zahlungen zwischen den Vermittlern auf dem gesamten Weg vom Sender zum Empfänger der ursprünglichen Zahlung. Eine bedingte Zahlung bedeutet in diesem Fall, dass der Intermediär nur dann eine Zahlung vom Absender erhält, wenn er seinen unmittelbaren rechten Empfänger bezahlt. Wenn derartige bedingte Zahlungen auf dem gesamten Zahlungsweg erfolgreich sind, bedeutet dies, dass der Absender an den vorgesehenen Empfänger gezahlt hat. Wir entwickeln neue Werkzeuge und Techniken sowohl auf der kryptografischen als auch auf der Transaktionsschicht, nämlich *lockable signatures* und *local 3-party channels*. Zusammen helfen uns diese neuen Werkzeuge, ein sicheres PCN-Protokoll zu erreichen, ohne auf die spezifischen algebraischen Eigenschaften des zugrundeliegenden Signaturschemas angewiesen zu sein, und stellen sicher, dass kein feindlicher Benutzer im Zahlungspfad die Münzen eines ehrlichen Benutzers stiehlt. Wir zeigen die Nützlichkeit unseres generischen Protokolls, indem wir es für den Fall von *Boneh-Lynn-Shacham (BLS)*-Signaturen [BLS01],das mehrere nützliche Eigenschaften bietet, wenn es für die Transaktionsauthentifizierung verwendet wird, wie Kompaktheit, Aggregation, etc., effizient instanziieren.

2.2 Universal Cross-Currency Exchange

Der Währungsumtausch ist die Grundlage wirtschaftlicher Aktivitäten und Kryptowährungen bilden keine Ausnahme. In traditionellen Umgebungen gehen zwei Nutzer, Alice und Bob, die ihre Währung tauschen möchten, zu einem Tauschdienst, der die ursprüngliche Währung akzeptiert und die gewünschte Währung zurückgibt. In diesem Fall ist der Tauschdienst eine vertrauenswürdige Einrichtung wie eine Bank. In der Kryptowährungsumgebung wollen wir uns natürlich nicht auf eine vertrauenswürdige Einrichtung verlassen. Es stellt sich also die Frage, wie Benutzer Währungen auf vertrauenswürdige Weise tauschen können.

Atomic-Swap-Protokolle [Wh, Su] wurden entwickelt, um zwei Nutzern, die Währungen tauschen möchten, die Möglichkeit zu geben, ihre Münzen untereinander zu tauschen, ohne zusätzliche vertrauenswürdige Instanzen einzubeziehen. Diese Protokolle basieren jedoch auf dem HTLC-Skript und leiden daher unter Fungibilitäts- und Kostenproblemen. Darüber hinaus sind sie mit verschiedenen Währungen wie Monero, Zcash usw., die keine HTLC-Skripte unterstützen, nicht kompatibel. Uns war kein atomares Swap-Protokoll bekannt, das keine speziellen Skripte der beteiligten Währungen und keine zusätzlichen Vertrauensannahmen erforderte.

In dieser Arbeit [Th22] entwerfen wir das erste universelle atomare Tauschprotokoll, das es Nutzern ermöglicht, eine beliebige Anzahl ihrer Coins zwischen beliebigen Währungen zu tauschen, vorausgesetzt, die Währungen unterstützen das Signaturprüfungsskript und das `locktime`-Skript. Wir können die Anforderung des `locktime`-Skripts durch andere Techniken, die wir weiter unten beschreiben, loswerden. Uns ist keine frühere Arbeit bekannt, die einen solchen universellen Tausch unterstützt, ohne dass komplexe Skripte, hohe Kosten oder ein vertrauenswürdiger Austauschdienst erforderlich sind. Darüber hinaus befassten sich frühere Arbeiten nur mit dem Tausch einzelner Vermögenswerte, bei dem zwei Benutzer jeweils eine Münze tauschen. Unser Protokoll hingegen kann mit mehreren Vermögenswerten umgehen, d. h. n Münzen eines Benutzers können atomar mit $\tilde{n}$ Münzen eines anderen Benutzers getauscht werden, wobei diese Münzen von jeder beliebigen Währung stammen können.

Als neue Techniken führen wir *multi-lock signatures* ein, die als eine Verallgemeinerung von *lockable signatures* angesehen werden können, die sicherstellen, dass den Benutzern während des Tauschs ein Begriff von *atomicity* garantiert wird. Genauer gesagt besagt die Atomizitätseigenschaft, dass, wenn mindestens eine der Münzen getauscht wird, die Benutzer alle verbleibenden Münzen innerhalb einer angemessenen Zeit tauschen können. Man kann unser Protokoll als Erleichterung von bedingten Zahlungen betrachten, bei denen ein Nutzer nur dann zahlt, wenn er bezahlt wird. Wir glauben, dass dieses Konzept breitere Anwendungen hat und von unabhängigem Interesse ist. Außerdem zeigen wir ein hocheffizientes Protokoll für Signaturverfahren wie ECDSA und Schnorr, die heute in den meisten Kryptowährungen als Authentifizierungsmechanismus verwendet werden.

2.3 Zeitlich begrenzte Zahlungen ohne Skripte

Wir konzentrieren uns dann auf die Verwendung von `locktime`-Skripten zur Implementierung von zeitlich begrenzten Zahlungen, bei denen eine Zahlung erst nach Ablauf einer bestimmten Zeit erfolgreich ist (d.h. das Skript erfüllt). Solche zeitgesteuerten Zahlungen helfen bei der Implementierung von Zahlungsströmen, bei denen eine Zahlung von Benutzer A an Benutzer B zur Zeit **T** abläuft, weil es eine zeitgesteuerte Zahlung (mit der Zeit **T**) gibt, die die Münzen von Benutzer B zurück an Benutzer A überträgt. Dies ist für viele Protokolle wie PC, PCN und atomare Swap-Protokolle von entscheidender Bedeutung, bei denen Benutzer gemeinsam Gelder in einer Adresse sperren und die Garantie haben wollen, dass selbst wenn einer der Benutzer für eine lange Zeit offline geht, die Gelder nach einer bestimmten Zeit **T** von dieser Adresse zurückgeholt werden können.

In dieser Arbeit [Th22] schlagen wir ein neues kryptographisches Primitiv namens *verifiable timed signatures (VTS)* vor, das bisher nur informell untersucht wurde. Mit diesem neuen Primitiv können wir den beabsichtigten Effekt einer zeitgesteuerten Zahlung erreichen, ohne ein `locktime`-Skript zu verwenden. Intuitiv lässt ein VTS einen Benutzer eine Signatur an eine Nachricht binden, so dass jeder andere Benutzer mit dieser Bindung die Signatur nach einer vorgegebenen Zeit **T** erhalten kann.

Um unsere *verifiable timed signatures* zu konstruieren, stützen wir uns auf *timelock puzzles*, das von Rivest, Shamir und Wagner [RSW96] erfunden wurde. Ein timelock puzzle erlaubt es einem Benutzer, eine Lösung in ein Puzzle einzubetten, so dass das Puzzle erst nach einer Zeitspanne von **T** gelöst werden kann, die seit der Erstellung des Puzzles vergangen ist. Die entscheidende Sicherheitsgarantie ist, dass ein Angreifer, egal wie viel parallele Rechenleistung er hat, die eingebettete Lösung nicht vor der Zeit **T** erfahren kann. Sie stellen eine praktische timelock puzzle Konstruktion vor, die auf der *sequential squaring assumption* in einer RSA-Gruppe basiert. Im Falle von VTS bettet der Committer, der eine Signatur kennt, diese in ein timelock puzzle ein und setzt die Commitment als das resultierende Puzzle. Außerdem fügt der Committer einen Beweis bei, dass das Rätsel tatsächlich eine gültige Signatur enthält. Die Eigenschaft, die wir von dem Beweis verlangen, ist die von *Zero-Knowledge*, die besagt, dass der Beweis keine Informationen über die Signatur im Inneren des Puzzles preisgibt, sondern nur, dass das, was sich im Inneren des Puzzles befindet, eine gültige Signatur ist. Wir wollen auch *soundness*, die sicherstellt, dass bei einer Verpflichtung und einem gültigen Beweis die Lösung des Rätsels keine ungültige Signatur ergibt.

Wir schlagen außerdem hocheffiziente Konstruktionen für dieses neue Primitiv für Fälle vor, in denen der Benutzer einige der am häufigsten verwendeten Standardsignaturverfahren wie ECDSA, Schnorr und BLS verwendet. Dies deckt bereits eine große Anzahl von Währungen ab, die heute existieren. In unseren Konstruktionen geben wir eine effiziente Instanziierung von timelock puzzles und den zugehörigen Beweis an, so dass sie eng miteinander gekoppelt sind. Genauer gesagt nutzen wir die algebraische Struktur der oben genannten Signaturschemata zusammen mit der Verwendung des kürzlich eingeführten *homomorphic timelock puzzles* [MT19], das uns eine hohe Lösungseffizienz bietet. Um eine zeitlich begrenzte Zahlung zu erreichen, erzeugt ein Benutzer eine VTS-Verpflichtung für die Signatur der Transaktion. Nun kann eine gültige Signatur der Transaktion erst nach Ablauf der Zeit **T** auf der Blockchain validiert werden. Dies imitiert die Wirkung des Skripts `locktime`, wenn es die Transaktion wäre. Wir erörtern auch, wie unser VTS und seine Erweiterungen dazu beitragen, dass das `locktime`-Skript sowohl in unserem PCN- als auch in unserem Atomic-Swap-Protokoll überflüssig wird.

Formale Sicherheitsgarantien. Alle in dieser Dissertation vorgeschlagenen neuen kryptographischen Primitive und Protokolle werden durch präzise formale kryptographische Sicherheitsdefinitionen unterstützt. Wir modellieren unsere Definitionen in einem möglichst starken Rahmen, der es erlaubt, verschiedene sichere Bausteine in der Praxis sicher zusammenzusetzen. Darüber hinaus bieten wir eine formale Sicherheitsanalyse unserer Primitive und Protokolle unter Verwendung kryptographischer Standardreduktionsverfahren.

2.4 Praktische Implikationen

Wie oben angedeutet ist *Lightning Network* ein praktisch eingesetztes PCN-System auf Bitcoin. Der Einsatz beruht auf den HTLC-Skripten und es gibt Entwicklungen, um diese Anforderung zu entfernen und sich nur auf Skripte zur Signaturprüfung zu stützen, wobei das Signaturschema entweder Schnorr-Signaturen oder ECDSA-Signaturen sind. Es gibt jedoch auch andere Währungen, die BLS-Signaturen und Post-Quantum-Signaturen zur Authentifizierung verwenden wollen, und unsere PCN-Protokolle sind für diese Entwickler von Interesse.

Das Gleiche gilt für atomare Swap-Protokolle, bei denen die derzeit eingesetzten Protokolle auf HTLC-Skripten basieren. Unsere atomaren Swap-Protokolle können nun von Nutzern verwendet werden, die eine beliebige Währung verwenden und sogar den Austausch mehrerer Münzen unterstützen. Unser effizientes Protokoll für den Fall von Schnorr- und ECDSA-Signaturen kann in allen wichtigen Währungen eingesetzt werden, da dies die bekanntesten Signaturverfahren sind, die derzeit zur Authentifizierung verwendet werden.

Unsere VTS-Konstruktionen sind auf die oben genannten prominenten Signaturverfahren zugeschnitten und können daher in allen wichtigen Währungen eingesetzt werden. Um diese exemplarisch zu zeigen wurde das erste Zahlungskanalprotokoll für die größte datenschutzfreundliche Art von Währung, zu der auch Monero gehört, entwickelt, das auf einer einfachen Erweiterung unserer VTS-Konstruktionen basiert. Die Entwickler der Währung arbeiten aktiv an der Einführung des Systems in Monero.

Alle oben genannten Konstruktionen wandeln nicht nur mehr Währungen tatsächlich nützliche Zahlungsprotokolle, sie verbessern auch die Privatsphäre und die Kosten der beteiligten Nutzer.

3 Zukünftige Arbeit

Es wurden bereits Folgearbeiten begonnen, bei denen Techniken aus dieser Arbeit zur Entwicklung neuer Zahlungsprotokolle verwendet wurden. Zum Beispiel wurde in [Th20] eine Erweiterung unseres VTS vorgeschlagen, die skalierbare Zahlungslösungen für Monero ermöglicht. Eine andere Arbeit [Au21] schlägt ein neues bidirektionales payment channel protokoll vor, das VTS verwendet und mit einer breiteren Klasse von Währungen kompatibel ist, ohne dass die Benutzer ständig online sein müssen und ohne dass eine Unterstützung durch Dritte erforderlich ist.

Wir arbeiten derzeit an der Erweiterung unseres payment channel network-Protokolls, damit es zusammen mit anderen Zahlungsparadigmen verwendet werden kann, wie z.B. einem *payment channel hub*, über den Zahlungen mithilfe eines Servers geleitet werden, der Zahlungsrouting als Dienstleistung anbietet. Durch die Kombination dieser beiden Paradigmen können wir allgemeinere Zahlungsverarbeitungsszenarien für Benutzer erfassen.

4 Fazit

Zusammenfassend lässt sich sagen, dass wir in dieser Dissertation neue kryptografische Sperrmechanismen untersuchen und vorschlagen, die schnelle, sichere und private Zahlungen in verschiedenen Umgebungen ermöglichen, die mit einer breiten Klasse von Blockchain-basierten Kryptowährungen kompatibel sind. Ein erster Schritt zur Erreichung dieses Ziels besteht darin, dass unsere kryptografischen Sperren nur minimal von der Währung selbst und ihren Skripting-Funktionen abhängig sind. Dies hat zur Folge, dass Währungsentwickler keine speziellen Skripting-Funktionen mehr einbauen müssen, um solche Zahlungen zu unterstützen, da unsere Sperren diese mit deutlich besserer Sicherheit und Privatsphäre ermöglichen.

Literaturverzeichnis

[Au21] Aumayr, Lukas; Thyagarajan, Sri AravindaKrishnan; Malavolta, Giulio; Monero-Sánchez, Pedro; Maffei, Matteo: Sleepy Channels: Bitcoin-Compatible Bi-directional Payment Channels without Watchtowers. Cryptology ePrint Archive, 2021.

[BLS01] Boneh, Dan; Lynn, Ben; Shacham, Hovav: Short Signatures from the Weil Pairing. In (Boyd, Colin, Hrsg.): Advances in Cryptology — ASIACRYPT 2001. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 514–532, 2001.

[La19] Lai, Russell WF; Ronge, Viktoria; Ruffing, Tim; Schröder, Dominique; Thyagarajan, Sri Aravinda Krishnan; Wang, Jiafan: Omniring: Scaling private payments without trusted setup. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. S. 31–48, 2019.

[Ma19] Malavolta, Giulio; Moreno-Sanchez, Pedro; Schneidewind, Clara; Kate, Aniket; Maffei, Matteo: Anonymous Multi-Hop Locks for Blockchain Scalability and Interoperability. Proceedings 2019 Network and Distributed System Security Symposium, 2019.

[MT19] Malavolta, Giulio; Thyagarajan, Sri Aravinda Krishnan: Homomorphic time-lock puzzles and applications. In: Annual International Cryptology Conference. Springer, S. 620–649, 2019.

[Na08] Nakamoto, Satoshi: , Bitcoin: A peer-to-peer electronic cash system, 2008.

[PD16] Poon, Joseph; Dryja, Thaddeus: , The bitcoin lightning network: Scalable off-chain instant payments, 2016.

[RSW96] Rivest, Ronald L; Shamir, Adi; Wagner, David A: Time-lock puzzles and timed-release crypto. 1996.

[Sa14] Sasson, Eli Ben; Chiesa, Alessandro; Garman, Christina; Green, Matthew; Miers, Ian; Tromer, Eran; Virza, Madars: Zerocash: Decentralized anonymous payments from bitcoin. In: 2014 IEEE Symposium on Security and Privacy. IEEE, S. 459–474, 2014.

[Su] Submarine Swap in Lightning Network. https://wiki.ion.radar.tech/tech/research/submarine-swap.

[Th20] Thyagarajan, Sri Aravinda Krishnan; Malavolta, Giulio; Schmidt, Fritz; Schröder, Dominique: PayMo: Payment Channels For Monero. IACR Cryptol. ePrint Arch., 2020:1441, 2020.

[Th22] Thyagarajan, Sri AravindaKrishnan: Cryptographic Locks for Scriptless Cryptocurrency Payments. doctoralthesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2022. Link: https://opus4.kobv.de/opus4-fau/frontdoor/index/index/docId/17993.

[Wh] What is atomic swap and how to implement it. https://www.axiomadev.com/blog/what-is-atomic-swap-and-how-to-implement-it/.

[Wo14] Wood, Gavin et al.: Ethereum: A secure decentralised generalised transaction ledger. Ethereum project yellow paper, 151(2014):1–32, 2014.

Sri AravindaKrishnan Thyagarajan wurde am 21. Dezember 1993 in Tamil Nadu, Indien, geboren. Er absolvierte sein Bachelorstudium (2011-2015) in Informatik und Ingenieurwesen am NIT Trichy Indien. Anschließend absolvierte er seinen Master (2015-2016) in Informatik an der Universität des Saarlandes, Deutschland. Er schloss sein PhD. (2016-2021) mit *Summa Cum Laude* in Informatik an der Friedrich-Alexander-Universität Erlangen-Nürnberg, Deutschland, unter der Leitung von Prof. Dominique Schröder. Derzeit ist er als Postdoktorand an der Carnegie Mellon University, USA, und NTT Research, USA, tätig. Er ist spezialisiert auf angewandte Kryptographie mit Anwendungen in den Bereichen Cloud-Speicher, Mehrparteienberechnungen, Blockchain und Kryptowährungen. Er hat mehrere seiner Arbeiten auf verschiedenen hochrangigen Konferenzen wie IEEE Security & Privacy, ACM CCS, CRYPTO, etc. veröffentlicht. Sein Ziel ist es, neue kryptographische Primitive und Protokolle mit formalen Sicherheitsgarantien und realen Anwendungen zu entwerfen und zu entwickeln.

Generative Deep-Learning-Modelle für die automatische Analyse und Synthese von medizinischen Bilddaten mit pathologischen Strukturen[1]

Hristina Uzunova[2]

Abstract: Deep-learning-basierte Algorithmen haben sich im Bereich der medizinischen Bildverarbeitung als besonders geeignet erwiesen, allerdings benötigen diese eine große Trainingsdatenmengen mit gegebenen Expertenannotationen. Die Erstellung solcher annotierter Datensätze für Bilddaten mit vorhandenen pathologischen Strukturen gilt als besonders herausfordernd, da die Variabilität der Pathologien verglichen mit normalen anatomischen Strukturen enorm ist. In dieser Arbeit werden deep-learning-basierte generative Modelle eingesetzt und weiterentwickelt, um die Herausforderungen von pathologischen Strukturen zu bewältigen. Einerseits wird ein variationeller Autoenocoder für die unüberwachte Detektion von Pathologien eingesetzt, andererseits werden Ansätze basierend auf GANs (*generative adversarial networks*) entwickelt, um realistische künstliche annotierte Bilder mit pathologischen Strukturen zu generieren. Weiterhin können die vorgestellten Ansätze für die Verbesserung der Bildregistrierung und Segmentierung von Bildern mit Pathologien eingesetzt werden.

1 Einleitung

Medizinische Bildverarbeitungsmethoden, insbesondere KI-basierte Ansätze, haben in den letzten Jahren an Popularität gewonnen, da sie das Potential haben, die tägliche klinische Routine zu erleichtern, indem sie zeitaufwändige und fehleranfällige Prozesse automatisieren. Eine der größten Herausforderungen für automatische medizinische Bildanalyseverfahren ist die enorme Vielfalt medizinischer Bilder. Durch die rasante Entwicklung von Deep-Learning-Ansätzen wird diese Variabilität aufgrund der flexiblen und nichtlinearen Natur von neuronalen Netzen, die typischerweise die Datenvariabilität aus einem annotierten Datensatz erlernen, immer besser beherrschbar. Die Verarbeitung medizinischer Bilder, die pathologische Strukturen enthalten, bleibt jedoch auch unter Berücksichtigung der neuesten Entwicklungen eine Herausforderung. Einer der Gründe dafür liegt in der diffusen Natur pathologischer Strukturen. Um ihre Variabilität zu erfassen, benötigen neuronale Netze daher eine große Menge an annotierten Bildern. Abgesehen von der Tatsache, dass solche Annotationen nur selten verfügbar und schwer zu erzeugen sind, können neuronale Netze nur für die Verarbeitung einer sehr spezifischen pathologischen Struktur trainiert

[1] Englischer Titel der Dissertation: "Generative Deep Learning Models for the Automatic Analysis and Synthesis of Medical Image Data Featuring Pathological Structures"

[2] Universität zu Lübeck, Institut für medizinische Informatik, Ratzeburger Allee 160, 23562 Lübeck, Deutschland, hristina.uzunova@dfki.de

werden. Aus diesem Grund werden im Rahmen dieser Arbeit Strategien zur Reduktion der benötigten Datenmenge untersucht. Darüber hinaus beeinträchtigen pathologische Strukturen Algorithmen, die auf normale anatomische Strukturen abzielen. So stellen pathologische Daten eine besondere Herausforderung beispielsweise für Algorithmen der Bildregistrierung dar, da durch fehlende Korrespondenzen keine zuverlässige Bildausrichtung möglich ist. Ein weiteres Problem in diesem Zusammenhang ist die Verdeckung anatomischer Regionen. So kann z. B. ein Tumor im Gehirn die Ventrikel überlagern, so dass ein Algorithmus, der für die Segmentierung von Hirnventrikeln entwickelt wurde, keine guten Ergebnisse liefert. Des Weiteren, deformieren pathologische Strukturen häufig die umgebende Anatomie. Ein Beispiel dafür sind Tumore, die das umliegende Gewebe allmählich verdrängen.

Ausgehend von dieser Problematik werden in der vorgestellten Arbeit [Uz21] zwei Konzepte auf Basis generativer Modelle verfolgt. Zum einen wird eine unüberwachte Methode entwickelt, die keine annotierten Trainingsdaten zur Detektion von Pathologien benötigen. Die Haupterkenntnis für diesen Ansatz wurden in [UHE18; Uz19b] veröffentlicht. Zum anderen wird ein Verfahren zur direkten Generierung pathologischer Bilder mit Annotationen entworfen, um ein Training neuronaler Netze auf synthetischen Daten zu ermöglichen und die Anzahl der echten annotierten Trainingsdaten deutlich zu reduzieren. Vorwiegend wurden diese methodischen Entwicklungen in [UEH20a; UEH20b; Uz19a] publiziert.

2 Unüberwachte Pathologiedetektion mit variationellen Autoencodern

2.1 Methoden

Grundlagen Variationeller Autoencoder Autoencoder (AEs) sind neuronale Netze, die aus einem Encoder f_ϕ und einem Decoder g_θ bestehen, wobei ϕ und θ die trainierbaren Netzwerkparameter sind [KW14]. Der Encoder überführt eine Eingabe $\mathbf{x}$ (hier: ein Bild) in eine niedrigdimensionale latente Vektorrepräsentation $\mathbf{z}$ und der Decoder versucht aus $\mathbf{z}$ die Eingabe so gut wie möglich zu rekonstruieren, so dass $g_\theta(f_\phi(\mathbf{x})) \approx \mathbf{x}$. Variationelle Autoencoder (VAEs) können als eine Erweiterung von konventionellen Autoencodern angesehen werden, die den latenten Raum auf eine Normalverteilung begrenzen. Somit wird das Lernen der Identitätsfunktion verhindert und neue Bilder können generiert werden, indem Zufallsvektoren $\tilde{\mathbf{z}} \sim \mathcal{N}(0, \mathbf{I})$ aus einer Normalverteilung gezogen und durch den trainierten Decoder geschickt werden $\tilde{\mathbf{x}} = g_\theta(\tilde{\mathbf{z}})$.

Unüberwachte Pathologiedetektion Die Grundidee hinter der vorgestellten unüberwachten Pathologiedetektion ist, VAEs auf Daten von gesunden Patienten zu trainieren, so dass die gesunde Variabilität erlernt wird und pathologische Strukturen als Abweichungen der gelernten Norm detektiert werden können. Konkret werden hier zwei Grundannahmen getroffen: 1) Wenn ein VAE auf Daten gesunder Patienten aus einer Domäne trainiert wird, können solche Daten im Testfall auch gut rekonstruiert werden. Werden jedoch Daten mit pathologischen Strukturen als Eingabe benutzt, kann nur ein pseudo-gesundes Bild

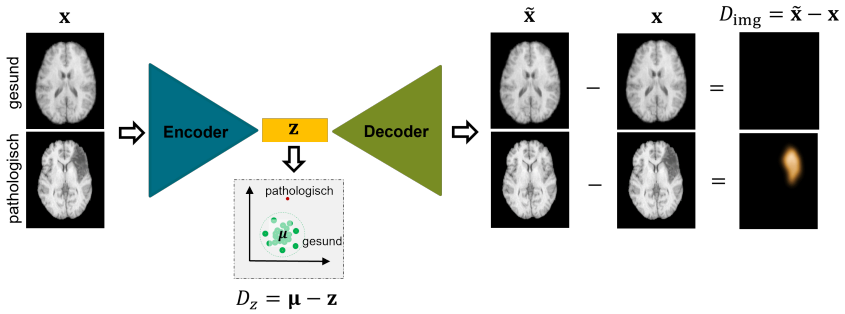
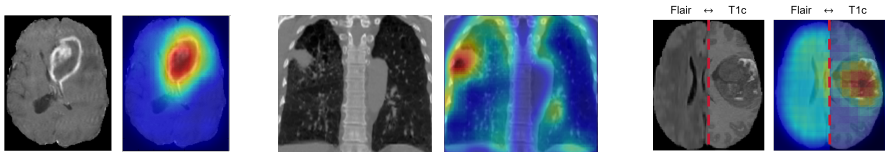


Abb. 1: Übersicht der zwei Pathologiedetektionsannahmen mit dem VAE.

rekonstruiert werden, so dass Pathologien als Rekonstruktionsfehler erkannt werden; 2) Die gelernte Normalverteilung im latenten Raum, bildet die normale Variabilität der Daten ab. Es wird also erwartet, dass pathologische Bilddaten weit entfernt von der gelernten Verteilung abgebildet werden (siehe Abb. 1). Eine multiplikative Kombination beider Annahmen wird in unseren Experimenten genutzt.

2.2 Experimente und Ergebnisse

Güte der Detektion pathologischer Strukturen Als Grundlage für die durchgeführten Experimente werden drei Datensätze, wie in Abb. 2 gezeigt, verwendet. Da für die Lungenbilder keine geeignete Grundwahrheit vorliegt, ist die Detektionsgüte hier nur qualitativ bewertet. Für die Gehirn-MRTs werden quantitative Metriken wie der Dice-Koeffizient und AUC (area under receiver operator curve) berechnet. Ergebnisse sind in Tab. 1 zu sehen und deuten auf eine gute Pathologiedetektion hin. Als Vergleich wird ein populäres GAN-basiertes Verfahren für die unüberwachte Detektion herangezogen: anoGAN [Sc19], wobei unsere vorgeschlagene Methode ähnliche Ergebnisse aufweist, jedoch eine wesentlich schnellere Inferenz und einen kleineren Speicherbedarf aufweist. Dies ermöglicht auch den Einsatz unserer Methode für 3D-Daten. Typisch für unüberwachte Verfahren kann auch hier keine pixelgenaue Segmentierung erreicht werden, trotzdem ist die Detektionsqualität ausreichend, um das Verfahren als Vorverarbeitungsschritt für weitere Algorithmen einzusetzen.



2D-Gehirn-MRT (T1c) 2D-Lungen-CT 3D-Gehirn-MRT (multimodal)

Abb. 2: Beispiele für genutzte Bilder und deren zugehörigen Pathologiedetektionen (Wärmebildkarte).

Metric	VAE (2D)	VAE (3D)	anoGAN (2D) [Sc19]
Dice ↑	0.55± 0.27	0.50±0.18	0.51± 0.28
AUC ↑	0.95	0.94	0.93

Tab. 1: Quantitative Ergebnisse der Detektionsgüte.

Unüberwachte Pathologiedetektion für die pathologische Bildregistrierung Ein häufig auftretendes Problem bei der Bildregistrierung sind fehlende Korrespondenzen, typischerweise verursacht von pathologischen Strukturen. Aus diesem Grund, wird in diesem Experiment angestrebt Vorinformationen über Pathologien in den Registrierungsprozess zu integrieren, wobei die oben beschriebene Pathologiedetektion verwendet wird. Als Basis wird ein etabliertes variationelles nicht-lineares Registrierungsverfahren gewählt [Eh15] und in eine atlas-basierte Registrierung eingesetzt. Die Information über mögliches pathologisches Gewebe wird über eine Gewichtsmaske so integriert, dass pathologische Regionen eine kleinere Rolle bei der Optimierung spielen und somit keine unplausiblen Deformationen in diesen Regionen entstehen. Für dieses Experiment werden 40 3D-Gehirn-MRTs [Sh08] mit synthetischen Gehirnläsionen verwendet. Quantitative Auswertungen der Registrierungsgüte werden in Form von gemittelten Dice-Koeffizienten anhand der vorgegebenen Segmentierungen durchgeführt. Ohne die Integration einer Gewichtsmaske wird ein Dice von 0.72 erreicht, mit der Gewichtsmaske ist der erreichte Dice von 0.73 statistisch signifikant höher ($p < 0.001$) und das Registrierungsdeformationsfeld wesentlich glatter, gemessen an der Standardabweichung der Jacobi-Determinante (0.37 gegenüber 0.55 ohne Gewichtsmaske).

Insgesamt zeigt dieses Verfahren Potential als Vorverarbeitungsschritt für weitere bildverarbeitende Algorithmen. Trotz zahlreicher Vorteile bietet diese Methode jedoch keine explizite Möglichkeit zur Pathologiemodellierung. Aus diesem Grund wird im weiteren Verlauf der Arbeit ein explizites GAN-basiertes Verfahren entwickelt.

3 GAN-basierte Synthese von hochaufgelösten medizinischen Bilddaten mit pathologischen Strukturen

3.1 Methoden

Grundlagen Generative Adversarial Networks Generative Adversarial Networks (GANs) sind neuronale Netze, die aus einem Generator g_θ und einem Diskriminator d_ξ bestehen, wobei θ und ξ die trainierbaren Netzwerkparameter sind [Go14]. Der Generator hat einen Zufallsvektor $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ als Eingabe und gibt ein Bild $\mathbf{x}_f = g_\theta(\mathbf{z})$ aus, das möglichst realistisch ist. Der Diskriminator wird trainiert, um zwischen echten $\mathbf{x}_r$ (*engl: real*) und generierten Bildern $\mathbf{x}_f$ (*engl: fake*) zu unterscheiden und gibt sein Feedback dem Generator zurück. Dieser versucht wiederrum, den Diskriminator zu täuschen, indem er möglichst

realistische Bilder generiert. So wird der so genannte Minimax-Algorithmus implementiert, indem Generator und Diskriminator sich alternierend fördern ihre Ergebnisse zu verbessern. Durch diese spieltheoretische Funktionsweise ermöglichen GANs die Synthese realistischer Bilddaten [Em21] und bilden die Grundlage der hier entwickelten Methodik.

Generierung pathologischer annotierter Datensätze Im Bereich der medizinischen Bildverarbeitung muss man sich häufig auf öffentlich verfügbare Datensätze als Trainingsgrundlage verlassen. Generell können die verfügbaren Datensätze jedoch in zwei Typen unterteilt werden: 1) Datensätze gesunder Subjekte mit gegebenen Segmentierungen relevanter anatomischer Strukturen [Sh08]; und 2) Bilder mit pathologischen Strukturen und deren Expertensegmentierung [Me15]. Also fehlen pathologische Daten mit Segmentierungen sowohl der anatomischen Regionen als auch der Pathologie. Ohne solche Daten können keine neuronalen Netze, die auf anatomische Strukturen in pathologischen Daten abzielen, trainiert werden: *Wie kann man ein Segmentierungsnetz für die Hirnventrikel in Tumorbilddaten trainieren?*; und eine Evaluation der Ergebnisse von Standardalgorithmen für anatomische Strukturen in pathologischen Daten ist unmöglich: *Wie gut kann ein auf gesunden Daten trainiertes Netz, die Hirnventrikel in Tumorbilddaten segmentieren?*

Motiviert dadurch wird in dieser Arbeit eine Pipeline zur Generierung realistischer Daten mit Annotationen anatomischer und pathologischer Strukturen entwickelt, hier gezeigt am Beispiel von Gehirn-MRTs mit segmentierten Hirntumoren, Ventrikeln und Caudate Nuclei. Folgende drei Schritte bilden die Pipeline (Abb. 3):

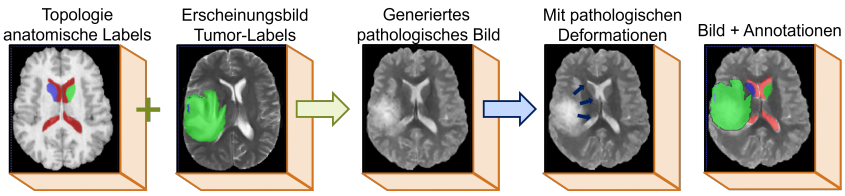


Abb. 3: GAN-basierte Pipeline für die Bildgenerierung. Topologieerhaltende Domain-Überführung für die Generierung von Bilddaten mit anatomischen und pathologischen Labels (■); 3D-Bildgenerierung mit MEGAN (■); und Simulation der pathologieinduzierten Gewebeverschiebung (■).

1) Topologieerhaltende Domain-Überführung: Die Grundidee hier ist eine Methode zu entwickeln, die das Erscheinungsbild von pathologischen Daten erlernt jedoch die Forminformation von normalanatomischen Daten behält, so dass gegebene anatomische Segmentierungen direkt übertragen werden können. Also wird eine topologieerhaltende Domain-Überführung von gesund auf pathologisch angestrebt. In der vorgelegten Arbeit wird diese mit den sog. *conditional* GANs (cGANs) etabliert, wobei diese auf intensitätsunabhängigen Topologieresktriktionen konditioniert werden und das Erscheinungsbild gegeben dieser bestimmen. Hier werden während des Trainings die Canny-Kantenbilder aus den pathologischen Bilddaten extrahiert und als Eingabe des GANs verwendet. Dieses versucht dann daraus das Originalbild zu rekonstruieren. Im Testfall können nun die Canny-Kantenbilder aus gesunden Daten extrahiert werden und in das trainierte Netz eingegeben, so dass die vorgegebene

Topologie erhalten bleibt, jedoch ein pathologisches Aussehen der Daten erreicht wird. Pathologische Strukturen werden explizit als Segmentierungsmasken überlagert, so dass eine leichte Kontrolle und Datenaugmentierung ermöglicht werden. Um den Domain-Shift zu verdeutlichen, werden hier unterschiedliche MRT-Sequenzen verwendet: T1 für gesund und T2 für pathologisch (Abb. 4).

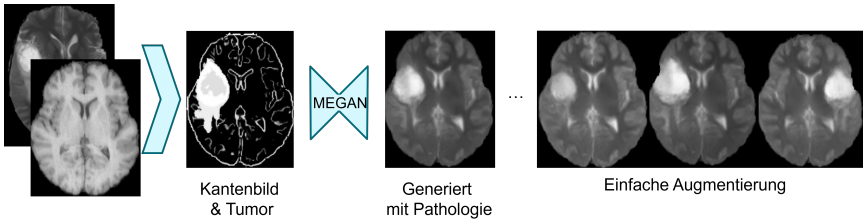


Abb. 4: Beispiel für generierte 3D-MRTs (Axial-Schichten) mit Gehirntumoren.

2) MEGAN: Speicher-effiziente Synthese großer Bildvolumina: Durch das anspruchsvolle Trainingsverfahren haben GANs einen außergewöhnlichen hohen (GPU-RAM) Speicherbedarf und somit eigenen sich diese in der Regel nicht für große hochaufgelöste 3D-Bilddaten, die für radiologische Untersuchungen unabdingbar sind. In unseren Untersuchungen mit gängigen GAN-Architekturen, wächst der Speicherbedarf kubisch in Abhängigkeit der isotrope Bildgröße. Aus diesem Grund entwickeln wir MEGAN (*memory-efficient GAN*) – ein patch-basiertes Multiskalen-Verfahren mit konstantem Speicherbedarf. Formal kann das Verfahren wie folgt beschrieben werden: gegeben sind die Kantenbilder $\mathbf{c}_0 \dots \mathbf{c}_n$ auf den Auflösungsskalen $s_0 \dots s_n$, die Ausgabebilder $\mathbf{x}_0 \dots \mathbf{x}_n$ werden von den Generatoren $g_0 \dots g_n$ und Diskriminatoren $d_0 \dots d_n$ synthetisiert, wobei die übliche GAN-Zielfunktion [Go14] wie folgt verändert werden kann:

$$\begin{aligned}
\min_{g_0}\max_{d_0} \mathcal{L}_{cGAN}(g_0,d_0) =& \mathbb{E}_{\mathbf{c}_0,\mathbf{x}_0}\left[\log d_0(\mathbf{c}_0,\mathbf{x}_0)\right] + \mathbb{E}_{\mathbf{c}_0,\mathbf{z}}\left[\log\left(1-d_0\Big(\mathbf{c}_0, g_0(\mathbf{c}_0,\mathbf{z})\Big)\right)\right] \\
\min_{g_i}\max_{d_i} \mathcal{L}_{cGAN}(g_i,d_i) =& \mathbb{E}_{\mathbf{c}_{p_i},\mathbf{x}_{p_i},\mathbf{x}_{p_{i-1}}}\left[\log d_i(\mathbf{c}_{p_i},\mathbf{x}_{p_{i-1}},\mathbf{x}_{p_i})\right] + \\
& \mathbb{E}_{\mathbf{c}_{p_i},\mathbf{x}_{p_{i-1}},\mathbf{z}}\left[\log\left(1-d_i\Big(\mathbf{c}_{p_i},\mathbf{x}_{p_{i-1}},g_i(\mathbf{c}_{p_i},\mathbf{x}_{p_{i-1}},\mathbf{z})\Big)\right)\right].
\end{aligned}$$

Hier sind $\mathbf{c}_{p_i}$ und $\mathbf{x}_{p_i}$ jeweils Patches aus $\mathbf{c}_i$ und $\mathbf{x}_i$, wo $i \in [1,n]$ die Auflösungsskala ist. In [UEH20b] wurde gezeigt, dass Bilder der außerordentlichen Größe von 512^3 Voxeln mit einer hohen Auflösung und ohne Artefakten generiert werden können, was die Grenzen der Verwendung von GAN-basierten Verfahren für medizinische Bilddaten wesentlich erweitert.

3) Simulation von pathologieinduzierter Gewebeverschiebung mit inversem probabilistischem Verfahren: Zwar können mit den oberen zwei Schritten hochauflösende realistische Bilder generiert werden (Abb. 4), jedoch können mittels einfacher Überlagerung der Pathologiemaske keine pathologieinduzierten Deformationen typisch vor allem für Gehirntumore

Moving	Fixed	Ventikel Dice ↑	Caudate N. Dice ↑
Real gesund	Real gesund	0.62	0.60
Synthetisch mit Tumor		0.62	0.63
Synthetisch mit Tumor+Def.		**0.47**	**0.47**
Real mit Tumor		**0.43**	**0.40**

Tab. 2: Ergebnisse des Evaluierungsexperiments. Gemittelte Dice-Koeffizienten für die anatomischen Strukturen Ventrikel und Caudate Nuclei. **Fett**-markierte Zahlen entsprechen statistisch gleichen Ergebnisse im doppelseitigen t-Test mit $p > 0.1$. Daraus folgt die erfolgreiche Approximation des Fehlers, wenn die Registrierung auf pathologischen Daten durchgeführt wird.

(sog. *tumor mass effect*) simuliert werden. Das Erlernen solcher komplexen Deformationen ist allerdings schwierig, da in den meisten Fällen keine longitudinalen Daten vorliegen, die die Entwicklung von gesund zu pathologisch erfassen. Aus diesem Grund wird in dieser Arbeit ein inverses Verfahren entwickelt, das aus einer pathologischen Form, das gesunde Äquivalent vorhersagt. Formal sei $\mathbf{s}_p \in \mathbb{R}^d$ die Form eines pathologisch deformierten Gehirns und $\mathbf{s}_h \in \mathbb{R}^d$ das jeweilige gesunde Äquivalent. Dann ist $f : \mathbb{R}^d \to \mathbb{R}^{d \times m}$ eine probabilistische Funktion (hier: probabilistisches U-Net [Ko18]) mit $f(\mathbf{s}_p) = \varphi$ und $\mathbf{s}_p \circ \varphi = \widetilde{\mathbf{s}}_h \approx \mathbf{s}_h$, wo φ ein Deformationsfeld ist und $m \in \{2, 3\}$ die Bilddimension beschreibt. Mit diesem Verfahren lassen sich realistische Deformationen simulieren, so dass sich pathologische Bilder wie in Abb. 5 gezeigt, generieren lassen.

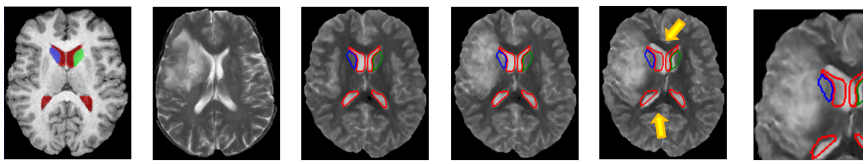


Abb. 5: Beispiel für generierte Bilder mit der vorgestellten Pipeline.

3.2 Experimente und Ergebnisse

Synthetische Bilder für die Evaluation von Standardalgorithmen Für die Evaluation werden die synthetisierten Bilder von einem auf gesunden Patientendaten vortrainierten Segmentierungsnetzwerk [Ya17] verwendet, wobei die Registrierungsgenauigkeit anhand der gegebenen anatomischen Labels berechnet wird (Dice-Koeffizient). Tab. 2 zeigt, dass die Ergebnisse für reale und synthetische pathologische Bilder statistisch gleich sind, was impliziert, dass die synthetischen Bilder als Approximation von realen Ergebnissen benutzt werden können. Im Gegensatz dazu würden Bilder gesunder Patienten oder ohne simulierten Pathologiedeformationen zu einer wesentlichen Überschätzung der Ergebnisse führen.

Experiment	Trainiert auf	Ventrikel Dice ↑	Caudate N. Dice ↑
Segmentierung	Synthetisch gesund	0.59	0.51
	Synthetisch mit Tumor	0.64	0.56
	Synthetisch mit Tumor+Def.	**0.70**	**0.57**
Registrierung	Keins	0.38	0.39
	Synthetisch gesund	0.52	0.52
	Synthetisch mit Tumor	0.50	0.49
	Synthetisch mit Tumor+Def.	**0.55**	**0.55**

Tab. 3: Ergebnisse des Trainingsexperiments. Gemittelte Dice-Koeffizienten für die anatomischen Strukturen Ventrikel und Caudate Nuclei. **Fett**-markierte Zahlen entsprechen statistisch signifikant besten Ergebnisse im doppelseitigen t-Test mit $p < 0.01$.

Synthetische Bilder für das Training neuronaler Netze Für dieses Experiment werden neuronale Netzwerke für die Segmentierung [RFB15] und Registrierung [Do15] auf synthetischen Daten trainiert und auf realen pathologischen Gehirn-MRTs evaluiert. Tab. 3 zeigt die Ergebnisse der Evaluation in Anbetracht der anatomischen Labels. Die synthetischen Bilder stellen sich als geeignete Trainingsdaten dar, wobei die Bedeutung der hinzugefügten pathologieinduzierten Deformation besonders herausgestellt wird, da nur so die besten Segmentierungs- und Registrierungsergebnisse erreicht werden.

4 Diskussion und Schlussfolgerungen

Zusammenfassend wird in der präsentierten Arbeit ein VAE-basiertes Verfahren vorgestellt, das für die unüberwachte Pathologiedetektion in medizinischen Bildern gut geeignet ist. Neben dem enormen Vorteil gegenüber überwachten Verfahren keine annotierten Bilddaten zum Training zu benötigen, ist dieses Verfahren auch mit weiteren unüberwachten Verfahren konkurrenzfähig. Trotz der fehlenden pixelgenauen Segmentierung, zeigt das Verfahren Potential als Vorverarbeitungsschritt für weitere bildverarbeitende Algorithmen eingesetzt zu werden. Trotz dieser Vorteile bietet die Methode keine direkte Möglichkeit zur Pathologiemodellierung. Aus diesem Grund wird ein weiteres explizites GAN-basiertes Verfahren entwickelt. Dieses ermöglicht es realistische Bilddaten mit Pathologien und Segmentierungen der pathologischen und anatomischen Regionen zu synthetisieren. Dies wird im ersten Schritt mit einer topologieerhaltenden Domain-Überführung erreicht. In einem weiteren Schritt wird das Verfahren MEGAN entwickelt, das es zum ersten Mal ermöglicht große und hochauflösende 3D-Volumen GAN-basiert zu generieren. Zusätzlich dazu wird eine Methode für die Simulation von pathologieinduzierten Gewebedeformationen entwickelt, die keine longitudinalen Daten benötigt. Die so generierten Daten zeigen in ausführlichen Experimenten ihre Eignung für die Evaluation und das Training von neuronalen Netzen. Zukünftig können weitere generative Modelle, z.B. deformierbare Autoencoder [UHE21], oder weitere Pathologietypen wie retinale Flüssigkeiten [Uz22] erforscht werden.

Literatur

[Do15] Dosovitskiy, A.; Fischer, P.; Ilg, E.; Hausser, P.; Hazirbas, C.; Golkov, V.; van der Smagt, P.; Cremers, D.; Brox, T.: Flownet: Learning Optical Flow with Convolutional Networks. In: Proceedings of the IEEE International Conference on Computer Vision - ICCV 2015. S. 2758–2766, 2015.

[Eh15] Ehrhardt, J.; Schmidt-Richberg, A.; Werner, R.; Handels, H.: Variational Registration - A Flexible Open-Source ITK Toolbox for Nonrigid Image Registration. In: Bildverarbeitung Für Die Medizin 2015. S. 209–214, 2015.

[Em21] Emami, H.; Aliabadi, M. M.; Dong, M.; Chinnam, R. B.: SPA-GAN: Spatial Attention GAN for Image-to-Image Translation. IEEE Transactions on Multimedia 23/1, S. 391–401, 2021.

[Go14] Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y.: Generative Adversarial Nets. In: Advances in Neural Information Processing Systems 27. S. 2672–2680, 2014.

[Ko18] Kohl, S.; Romera-Paredes, B.; Meyer, C.; De Fauw, J.; Ledsam, J. R.; Maier-Hein, K.; Eslami, S. M. A.; Jimenez Rezende, D.; Ronneberger, O.: A Probabilistic U-Net for Segmentation of Ambiguous Images. In: Advances in Neural Information Processing Systems 31. S. 6965–6975, 2018.

[KW14] Kingma, D.; Welling, M.: Auto-Encoding Variational Bayes. In: International Conference on Learning Representations. 2014.

[Me15] Menze, B. H. et al.: The Multimodal Brain Tumor Image Segmentation Benchmark (BRATS). IEEE Trans. Med. Imaging 34/10, S. 1993–2024, 2015.

[RFB15] Ronneberger, O.; Fischer, P.; Brox, T.: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. S. 234–241, 2015.

[Sc19] Schlegl, T.; Seeböck, P.; Waldstein, S. M.; Langs, G.; Schmidt-Erfurth, U.: F-AnoGAN: Fast Unsupervised Anomaly Detection with Generative Adversarial Networks. Medical Image Analysis 54/1, S. 30–44, 2019.

[Sh08] Shattuck, D. W.; Mirza, M.; Adisetiyo, V.; Hojatkashani, C.; Salamon, G.; Narr, K. L.; Poldrack, R. A.; Bilder, R. M.; Toga, A. W.: Construction of a 3D Probabilistic Atlas of Human Cortical Structures. NeuroImage 39/1, 2008.

[UEH20a] Uzunova, H.; Ehrhardt, J.; Handels, H.: Generation of Annotated Brain Tumor MRIs with Tumor-Induced Tissue Deformations for Training and Assessment of Neural Networks. In: Medical Image Computing and Computer Assisted Intervention – MICCAI 2020. 2020.

[UEH20b] Uzunova, H.; Ehrhardt, J.; Handels, H.: Memory-Efficient GAN-Based Domain Translation of High Resolution 3D Medical Images. Computerized Medical Imaging and Graphics 86/1, 2020.

[UHE18] Uzunova, H.; Handels, H.; Ehrhardt, J.: Unsupervised Pathology Detection in Medical Images Using Learning-Based Methods. In: Bildverarbeitung Für Die Medizin 2018. 2018.

[UHE21] Uzunova, H.; Handels, H.; Ehrhardt, J.: Guided Filter Regularization for Improved Disentanglement of Shape and Appearance in Diffeomorphic Autoencoders. In: Medical Imaging with Deep Learning. 2021.

[Uz19a] Uzunova, H.; Ehrhardt, J.; Jacob, F.; Frydrychowicz, A.; Handels, H.: Multi-Scale GANs for Memory-Efficient Generation of High Resolution Medical Images. In: Medical Image Computing and Computer Assisted Intervention – MICCAI 2019. 2019.

[Uz19b] Uzunova, H.; Schultz, S.; Handels, H.; Ehrhardt, J.: Unsupervised Pathology Detection in Medical Images Using Conditional Variational Autoencoders. International Journal of Computer Assisted Radiology and Surgery 14/3, S. 451–461, 2019.

[Uz21] Uzunova, H.: Generative Deep Learning Modelle für die automatische Analyse und Synthese von medizinischen Bilddaten mit pathologischen Strukturen, Dissertation, Universität zu Lübeck, 2021.

[Uz22] Uzunova, H.; Basso, L.; Ehrhardt, J.; Handels, H.: Synthesis of annotated pathological retinal OCT data with pathology-induced deformations, Accepted at SPIE Medical Imaging 2022: Image Processing, 2022.

[Ya17] Yang, X.; Kwitt, R.; Styner, M.; Niethammer, M.: Quicksilver: Fast Predictive Image Registration – A Deep Learning Approach. NeuroImage 158/1, S. 378–396, Sep. 2017.

Hristina Uzunova wurde am 6. Juli 1993 in Bulgarien geboren. Nach ihrem Abitur begann sie das Studium der Medizinischen Informatik an der Universität zu Lübeck im Jahr 2012. Im Jahr 2015 erfolgte der Bachelorabschluss zum Thema *Robuste affine Registrierung medizinischer Bilddaten mithilfe des RASL-Algorithmus* mit Auszeichnung und wurde mit dem Best-Bachelor-Preis von Philips prämiert. Im anschließenden Masterstudium vertiefte sie ihre Expertise im Bereich der medizinischen Bildverarbeitung und schloss mit einer Masterarbeit zum Thema *Detektion von Pathologien in medizinischen Bildern mit lernbasierten Verfahren* als Jahrgangsbeste ab. Ihre Masterarbeit wurde mit dem 1.Platz des conhIT-Nachwuchspreises geehrt. Nach vier Jahren wissenschaftlicher Arbeit am Institut für Medizinische Informatik in Lübeck schloss sie ihre Dissertation mit Bestnote ab. Im Rahmen ihrer Dissertation zum Thema *Generative Deep-Learning-Modelle für die automatische Analyse und Synthese von medizinischen Bilddaten mit pathologischen Strukturen* veröffentlichte sie 15 Originalarbeiten in hochrangigen Journalen und Tagungsbänden. Seit 2021 setzt Hristina Uzunova ihre wissenschaftliche Tätigkeit am Deutschen Forschungszentrum für Künstliche Intelligenz fort.

Erkennung und Analyse von Speicheranomalien in Sprachen mit automatischer Speicherverwaltung unter Nutzung von Trace-basierter Speicherüberwachung[1]

Markus Weninger[2]

Abstract: Moderne Programmiersprachen nutzen automatische Speicherbereinigung, um fehleranfällige manuelle Speicherverwaltung zu vermeiden. Dennoch können Anomalien wie Speicherlecks auftreten, die sich drastisch auf die Leistung einer Anwendung auswirken und sogar Abstürze herbeiführen können. Die meisten modernen Werkzeuge nutzen für ihre Speicheranalysen jedoch leider nur Speicherauszüge, d.h. sie inspizieren den Speicher nur an einem oder wenigen bestimmten Zeitpunkten. Diese bieten aber oft nicht genug Details, um zur Ursache des Problems vorzudringen. Unser Ansatz nutzt daher Traces, kontinuierliche Aufnahmen von Ereignissen wie beispielsweise Allokationen oder Speicherbereinigungsoperationen. Diese Arbeit zeigt, wie Traces genutzt werden können, um die (automatische) Speicherproblemerkennung und -analyse zu verbessern. Sie schlägt unter anderem Algorithmen zur Aufzeichnungsverarbeitung vor und führt neuartige Anomalieanalysen (z.B. die automatisierte Analyse des Wachstums von Datenstrukturen) sowie interaktive Visualisierungstechniken ein. Ferner untersucht sie, wie (unerfahrene) Benutzer sich bei der Speicheranalyse verhalten und wie Werkzeuge verbessert werden können, um diese Nutzer besser zu unterstützen und anzuleiten.

1 Motivation

In systemnahen Sprachen wie beispielsweise C liegt es in der Verantwortung des Programmierers, Allokationen und Deallokationen korrekt vorzunehmen. Während dies Flexibilität bietet und hochperformanten Code ermöglicht, kann diese manuelle Speicherverwaltung leicht zu unbeabsichtigten Speicherfehlern führen. Die häufigsten dieser Fehler sind auf Speicherlecks (vergessene `free()` Operationen) oder hängende Zeiger (Zugriffe auf bereits freigegebenen / nicht allokierten Speicher) zurückzuführen. Am besten kann dies wohl mit dem Mantra *Aus großer Kraft folgt große Verantwortung* zusammengefasst werden.

Um diesen Problemen entgegenzuwirken, nutzen automatisch speicherbereinigende Sprachen wie beispielsweise Java *Garbage Collection (GC)*, zu deutsch (lit.) *Müllabfuhr*. Während einer solchen werden Objekte, die nicht länger (indirekt) von Speicherwurzeln (engl. *GC roots*, wie beispielsweise statische Felder oder lokale Variablen) aus erreichbar sind, automatisch entfernt und ihr Speicher freigegeben. Leider ist die automatische Speicherbereinigung ebenso anfällig für mögliche Speicherprobleme. Diese können die

[1] Englischer Titel der Dissertation: Detection and Analysis of Memory Anomalies in Managed Languages Using Trace-Based Memory Monitoring

[2] Johannes Kepler Universität Linz, Institut für Systemsoftware, Altenbergerstraße 69, 4040 Linz, Österreich
markus.weninger@jku.at

Applikation verlangsamen, falls Entwickler achtlos mit Objektallokationen und der Speicherung von Objekten umgehen. Im schlimmsten Fall können Speicherlecks dazu führen, dass die Applikation nicht nur langsamer wird, sondern komplett abstürzt.

Speicherlecks in Sprachen mit GC treten auf wenn Objekte, die nicht mehr benötigt werden, durch Programmierfehler weiterhin von Speicherwurzeln aus erreichbar bleiben [MBR10]. Beispielsweise kann ein Programmierer vergessen, Objekte aus langlebigen Datenstrukturen zu entfernen, obwohl sie nicht mehr benötigt werden. Der Garbage Collector kann diese Objekte dann nicht freigeben – sie häufen sich also an und der Speicher läuft voll [XR13].

Eine weitere Speicheranomalie ist *hohe Speicherfluktation* (engl. *high memory churn*, auch *excessive dynamic allocations* [PH16; SW00] oder *high allocation density* [Du03]). Dieses Problem tritt auf, wenn Objekte (unnötigerweise) in hoher Frequenz allokiert werden, nur um sie kurz nach ihrer Erzeugung wieder freizugeben. Dies führt dazu, dass Laufzeit sowohl für die Allokation der Objekte selbst, als auch für die Speicherbereinigung ebendieser durch den Garbage Collector, aufgewendet werden muss. Beide Punkte beeinflussen die Performanz einer Applikation negativ.

Um dem entgegenzuwirken, ist es daher von entscheidender Bedeutung, Entwicklern intelligente Speicherwerkzeuge an die Hand zu geben, welche (semi-automatische) Analysen und Funktionen zum Aufspüren, Verstehen und Beheben von Speicheranomalien bieten.

2 Überblick

Da Speicherauszüge (engl. *heap dumps*) oft nicht genug Details bieten, um zur grundsätzlichen Ursache eines Speicherproblems vorzudringen, fokussiert sich diese Arbeit auf die Nutzung von Speicheraufzeichnungen (engl. *memory traces*) zur Anomalieerkennung und -analyse. In ihren Anfängen hatten Traces das Problem, dass der Aufwand, der betrieben werden musste, um diese aufzuzeichnen, oft die Laufzeit der analysierten Applikation um mehr als das 100-fache erhöhte [He06; RGM13; Xu13]. Dies machte Traces irrelevant für Einsätze in Realumgebungen. In den letzten Jahren wurde jedoch gezeigt, dass die Laufzeiterhöhung, welche durch die Aufzeichnung verursacht wird, auf wenige Prozent reduziert werden kann [LBM15; LBM16; Le16]. Ein Großteil der bestehenden Arbeiten in diesem Bereich (1) fokussiert sich entweder auf die (effiziente) Sammlung von Speicheraufzeichnungen, ohne jedoch neuartige Analysen basierend auf diesen Daten zu zeigen, oder (2) präsentiert stark problemspezifische Traceformate, welche für eine bestimmte Art der Speicheranalyse genutzt werden können, untersuchen aber nicht, ob diese auch für weitere Arten der Speicheranalyse genutzt werden könnten.

Diese Thesis beschäftigt sich aus diesem Grund mit der Nutzung von generellen Speicheraufzeichnungen (d.h., Speicheraufzeichnungen die sich nicht auf eine bestimmte Analyse fokussieren). Als Beispiel dienen jene Traces, die durch die AntTracks VM, einer virtuellen Maschine von Lengauer et al. [LBM15; LBM16; Le16] basierend auf der Java Hotspot

VM [Or21], produziert werden. Im Speziellen untersucht die Arbeit die Frage, wie Traces für Analysen in entwicklerorientierten Speicheranalysewerkzeugen genutzt werden können. Um dies zu untersuchen, fokussieren wir uns auf Fragen und Herausforderungen in Bezug auf die *Verarbeitung und Aufbereitung* von Traces, wie man basierend auf ihnen eine gemeinsame Datenbasis für verschiedenste Analysen und Visualisierungen schaffen kann, und wie solche Analysen und Visualisierungen aussehen können, um Entwickler bestmöglich beim *Erkennen, Analysieren und Beheben* von Speicheranomalien zu unterstützen. Weiters untersucht die Arbeit, wie sich (unerfahrene) Benutzer während der Speicheranalyse verhalten, auf welche Probleme und Hindernisse diese dabei stoßen und wie Speicherwerkzeuge im Allgemeinen verbessert werden können, um diese Nutzer besser zu leiten.

3 Themengebiete

Die Ergebnisse dieser Dissertation wurden in 13 Publikationen veröffentlicht, zwei davon in referierten Fachzeitschriften, die anderen auf internationalen Konferenzen. Wir verzichten auf Grund von Längenbeschränkungen auf Einzelreferenzen und verweisen auf die Gesamtdissertation [We21]. Die Arbeiten können in folgende Themen unterteilt werden.

3.1 Speicheraufzeichnungen und deren Verarbeitung

Wir präsentieren einen neuartigen Ansatz zur Speicherdatenaufbereitung basierend auf einem flexiblen Klassifikationssystem. In seinem Kern gruppiert das System Heapobjekte nach vom Nutzer bestimmten Eigenschaften. Diese Eigenschaften können beispielsweise der Objekttyp, die Allokationsstelle, der allokierende Thread, oder ein beliebiges anderes Nutzer-spezifiziertes Merkmal sein. Gruppiert man Heapobjekte nach mehreren Gesichtspunkten, indem man beispielsweise alle Objekte zuerst nach ihren Typen gruppiert und anschließend alle Objekte eines Typs nach deren Allokationsstellen, so resultiert dies in einem *Speicherbaum* (siehe Abb. 1). Wir zeigen, dass solche Speicherbäume als Datenquelle für diverse Analysen wie beispielsweise der Heapanalyse, der Datenstrukturwachstumsanalyse, als auch für verschiedenste Visualisierungen geeignet sind. Unsere Arbeit zur Heapobjektgruppierung wurde auf der *ICPE 2018* für den *Best Paper Award* nominiert.

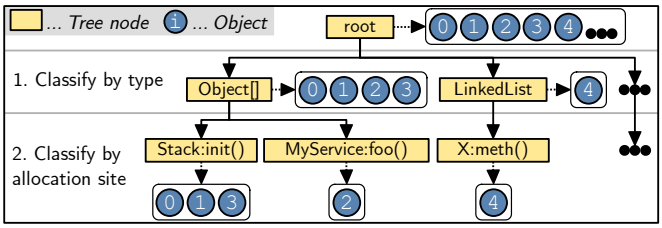
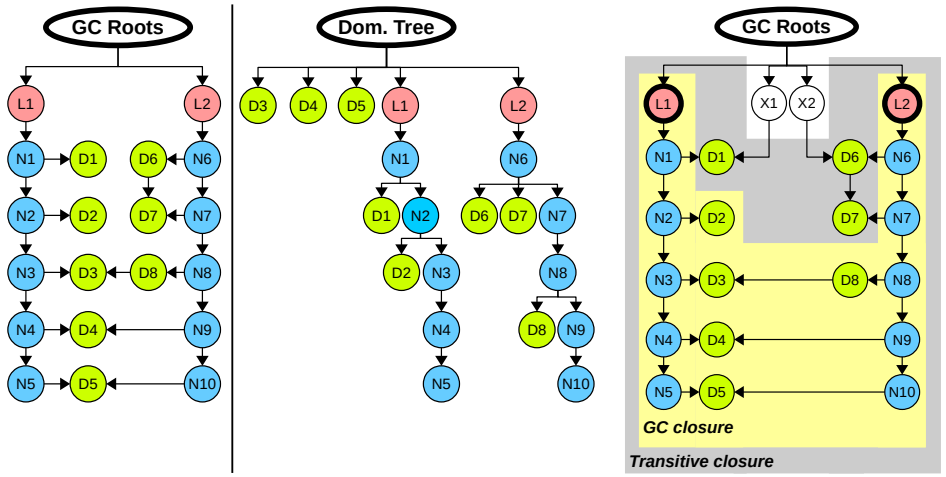


Abb. 1: Ein Speicherbaum, der zuerst alle Objekte nach deren Typ und anschließend nach deren Allokationsstelle gruppiert.

Die Thesis behandelt außerdem die Analyse von Objektreferenzen, d.h., welche Objekte welche anderen Objekte am Leben halten. Die meisten existierenden Ansätze greifen dazu auf die *Dominanzrelation* und *Dominanzbäume* zurück. Der gravierendste Nachteil dieser Ansätze ist jedoch, dass auf Basis der Dominanzrelation nur *Einzelobjektbesitz* analysiert werden kann, d.h., es werden nur jene Objekte erkannt, die durch *genau ein* anderes Objekt am Leben gehalten werden (engl. *single-object ownership*). Dies wird in Abb. 2a veranschaulicht, in dem zu sehen ist, dass für drei Datenobjekte `D3`, `D4` und `D5` mit Hilfe der Dominanzrelation kein Besitzer gefunden werden kann, da sie sowohl von der Liste `L1` als auch von `L2` am Leben gehalten werden. Unser Ansatz unterscheidet sich von Dominanzrelation-basierten Ansätzen darin, dass es uns möglich ist, Objekte und deren Besitzerverhältnis zu erkennen, auch wenn diese von *mehreren* anderen Objekten am Leben gehalten werden (engl. *multi-object ownership*). Dazu präsentieren wir Algorithmen zur Berechnung der *transitiven Hülle*, d.h., alle Objekte, die ausgehend von einem Objekt, oder aber auch einer Gruppe an Objekten, erreichbar sind, sowie Algorithmen zur Berechnung der *Garbage Collection Hülle*, d.h., jene Objekte, die ebenfalls vom Garbage Collector bereinigt werden können, sollte das Ausgangsobjekt, oder eine Gruppe an Ausgangsobjekten, zur Garbage Collection freigegeben werden. Abb. 2b zeigt beide Hüllenarten ausgehend von den beiden verketteten Listen `T1` und `T2`. Hier ist zu sehen, dass unsere Hüllenalgorithmen `D3`, `D4` und `D5` als Teil der Hüllen von `L1` und `L2` erkennt, da diese von beiden Listen am Leben gehalten werden. Solche Hüllen bieten wichtige Metriken zur Erkennung von verdächtigen Speichermustern, und unser Ansatz unterstützt die Analyse dieser Speichermuster somit sogar im Falle von multi-object ownership.



(a) Objektgraph von zwei einfach verketteten Listen `L1` und `L2` sowie deren Dominanzbaum.

(b) Transitive Hülle sowie Garbage Collection Hülle von $L1$ und $L2$.

Abb. 2: Objektgraph, Dominanzbaum, sowie die Hüllen zweier einfach verketteter Listen *L1* und *L2*.

3.2 Datenstrukturanalyse

Die zuvor erwähnten Hüllenalgorithmen sind besonders nützlich zur Analyse von Datenstrukturen und deren Entwicklung über Zeit, denn sie erlauben es uns, *Datenstrukturwachstum* zu erkennen. Wir entwickelten eine domänenspezifische Sprache, um zu beschreiben, welche Teile einer Datenstruktur *intern* sind (wie beispielsweise die Knoten einer `HashMap`) und welche Teile *extern* (wie beispielsweise Objekte, die als Schlüssel oder Werte in einer `HashMap` gespeichert werden). Kombiniert man diese Beschreibungen mit dem Wissen über die zeitliche Entwicklung der Hüllen jeder Datenstruktur, können Metriken abgeleitet werden, die es uns ermöglichen, Datenstrukturen basierend auf ihrer Wahrscheinlichkeit, an einem Speicherleck beteiligt zu sein, zu sortieren. Darüber hinaus können wir die Art von Lecks analysieren: ob es sich um ein datenstrukturinternes oder -externes Leck handelt und ob die anhäufenden Objekte von einem einzelnen Objekt am Leben gehalten werden (single-object ownership) oder von mehreren (multi-object ownership). Der gesamte Analyseprozess ist in Abb. 3 zusammengefasst.

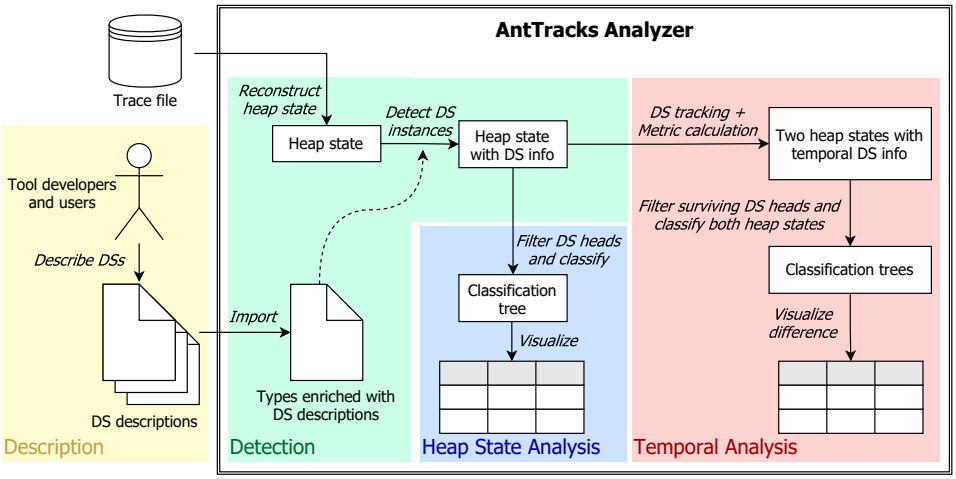


Abb. 3: Datenstrukturanalyseansatz aus 4 Stufen: (1) Beschreibung der Datenstrukturen mit unserer DSL, (2) Erkennung von Datenstrukturen in rekonsturierten Heapzuständen, (3) Heapanalyse, d.h., Datenstrukturanalyse zu einem bestimmten Zeitpunkt, und (4) Wachstumsanalyse, d.h., Erkennung verdächtig wachsender Datenstrukturen durch deren Verfolgung über Zeit.

3.3 Visualisierungen

Weiters präsentieren wir verschiedene Visualisierungen, die die Speicherentwicklung *über Zeit* darstellen. Dafür untersuchten wir bestehende Visualisierungsansätze [Sc11] und evaluierten sie in Bezug auf deren Anwendbarkeit und Nützlichkeit zur Darstellung von Speichermetriken, speziell der Entwicklung von Speicherbäumen über Zeit. Wir zeigen Anwendungen

und Adaptionen von traditionellen Zeitseriendiagrammen, zweidimensionalen *Baumvisualisierungen* sowie einer interaktiven dreidimensionalen *Speicherstadt*-Visualisierung. All diese Ansätze stellen neuartige Inspektions- und Interaktionsmechanismen zur Verfügung, die es erlauben, die Speicherentwicklung eines System leichter verständlich und greifbarer zu machen. Für jeden Ansatz präsentieren wir eine komplette Visualisierungpipeline [Li19], die Datenaufbereitung, Layouting, Darstellung, sowie mögliche Nutzerinteraktionen umfasst.

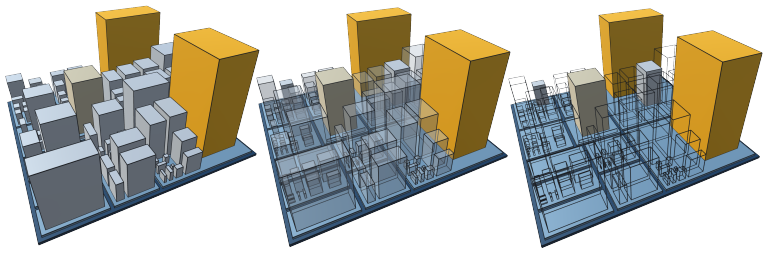


Abb. 4: Unsere Softwarestädte haben vielseitig modifizierbare Darstellungsmerkmale. Links: Jedes Gebäude ist voll deckend dargestellt. Mitte: Die fünf am stärksten wachsenden Gebäude sind voll deckend dargestellt, der Rest hat eine Deckkraft von 40%. Rechts: Die fünf am stärksten wachsenden Gebäude sind voll deckend dargestellt, die restlichen voll transparent (bis auf ihre Umrisse).

Beispielsweise adaptieren unsere Speicherstädte (Abb. 4) die Metapher der *Softwarestadt* [WL07]. Diese wurde in der Vergangenheit meist dazu genutzt, statische Metriken eines Softwaresystems (Klassenhierarchien, etc.) zu visualisieren. Wir erweitern diesen Ansatz um die Darstellung des *dynamischen Speicherverhaltens* einer Applikation. Gruppen von Heapobjekten (beispielsweise Objekte mit dem selben Objekttyp die in der selben Methode allokiert wurden) werden als Gebäude dargestellt, welche wiederum in Distrikten (beispielsweise alle Gebäude des selben Objekttyps) angeordnet sind. Die Größe eines Gebäudes entspricht dabei der Anzahl an Heapobjekten die es repräsentiert. Indem wir die Stadt kontinuierlich aktualisieren (entweder manuell durch den Nutzer oder durch das Abspielen einer automatischen Animation), erzeugen wir das Gefühl einer sich entwickelnden Stadt, in der wachsende Gebäude wachsende Objektgruppen im Heap darstellen. Durch die Nutzung von Farbe und Deckkraft lenken wir die Aufmerksamkeit der Nutzer noch weiter auf bestimmte Gebäude (zum Beispiel jene mit starkem Wachstum), mit denen dann interagiert werden kann, um diese genauer zu untersuchen.

Beide Arbeiten der VISSOFT 2020 (3D Softwarestadt, siehe Abb. 4) sowie der STAG 2020 (2D Baumvisualisierung, siehe Abb. 5) wurden mit dem *Best Paper Award* ausgezeichnet.

3.4 Speicherfluktationsanalyse

Wir haben im Zuge dieser Thesis nicht nur Speicherlecks, die wohl häufigsten Speicheranomalien, untersucht, sondern auch andere Speicherprobleme. Beispielsweise präsentieren wir eine Technik, die Nutzer automatisch auf Zeitfenster mit intensiver Speicherfluktation

hinweist. In solchen Zeitfenstern wird in kurzer Zeit eine große Menge an Objekten allokiert und kurz darauf wieder freigegeben. Wir zeigen, wie man für jedes Heapobjekt seinen *Geburtszeitpunkt* und *Bereinigungszeitpunkt* aus Speichertraces rekonstruieren kann, um daraus die *Lebensspanne* der Objekte zu berechnen, d.h., wie viele Garbage Collections jedes Objekt überlebt hat, bevor es vom GC wieder freigegeben wurde. Wir nutzen diese Informationen, um den Nutzer zu problematischen Stellen im Code zu leiten, um dort häufig allokierte Objekte genauer auf ihre Notwendigkeit hin zu untersuchen.

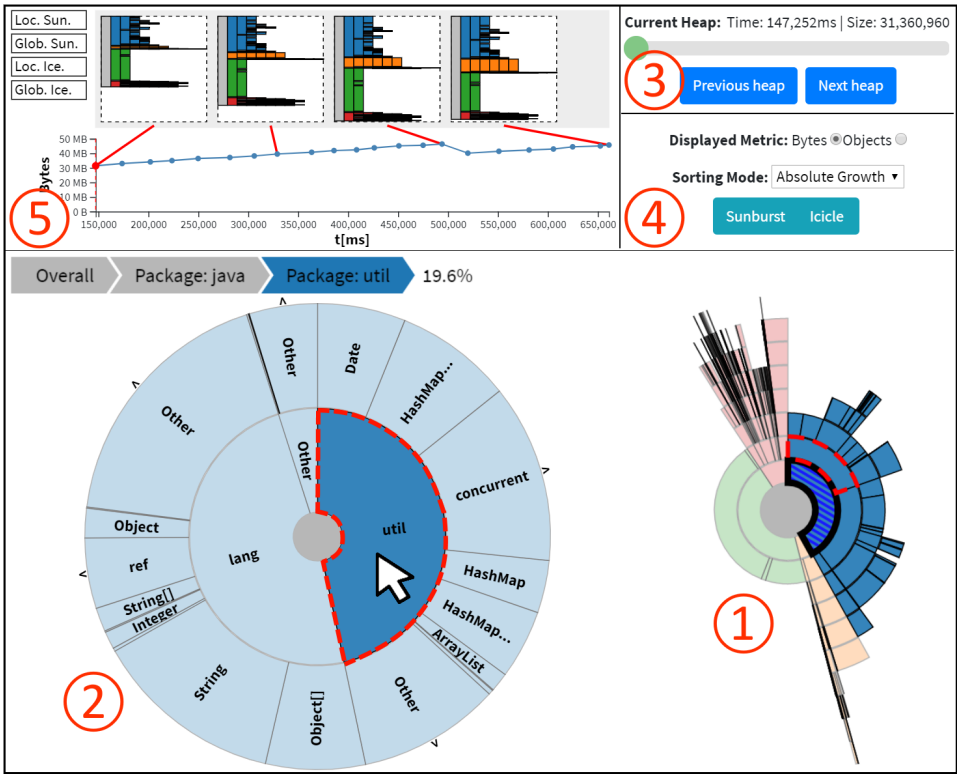


Abb. 5: Übersicht über unser Speicherbaum-Visualisierungstool. (1) und (2) zeigen eine *Zeitreisen-basierte* Visualisierung, wobei (1) den kompletten Baum mit allen Zweigen und Leveln zu einem bestimmten Zeitpunkt zeigt und (2) einen Teilausschnitt dieses Baumes zeigt, dessen Wurzel durch Klicken auf ein Baumsegment gewechselt werden kann. Beide Visualisierungen sind synchronisiert. Beispielsweise ist das Baumsegment, über dem sich der Mauszeiger befindet, sowohl in (1) als auch in (2) hervorgehoben (rote Umrandung). (3) zeigt die Oberfläche zur Zeitkontrolle und (4) bietet verschiedene Visualisierungsoptionen. Daneben befindet sich die (5) *Zeitlinien-basierte* Visualisierung, welche mehrere Bäume an verschiedenen Zeitpunkten nebeneinander anzeigt.

3.5 Anwenderunterstützung und Nutzerverhalten

Wir haben eine Nutzerstudie sowie strukturierte kognitive Durchgänge (engl. *structured cognitive walkthroughs* [BG03]) durchgeführt, um den Nutzen unserer Techniken zu überprüfen sowie ein besseres Verständnis dafür zu erlangen, wie unerfahrene Benutzer Speicherwerkzeuge und deren Ananlysemöglichkeiten nutzen. Basierend auf den Resultaten haben wir Empfehlungen ausgearbeitet, welche Speicherwerkzeugentwicklern helfen sollen, bestehende System zu verbessern und neue Funktionen zu implementieren.

Diesen Vorschlägen folgend haben wir selbst ein verbessertes Nutzerleitsystem in unsere Speicheranalysetechniken eingebaut. Wir entwickelten Algorithmen, die es ermöglichen, automatisch Muster in der Speicherauslastung einer Applikation zu erkennen, welche auf Speicheranomalien wie Speicherlecks oder Speicherfluktationen hinweisen. Wenn ein Zeitfenster mit einem solchen Muster erkannt wird, wird es hervorgehoben, um die Aufmerksamkeit des Nutzers darauf zu lenken.

Basierend auf dieser ersten Erfahrung haben wir ein Konzept zur Nutzerunterstützung namens *Geleitete Exploration (GE)* (engl. *guided exploration*) entworfen. Dieses Konzept dreht sich um vier Leitoperationen, die von Werkzeugen zur Verfügung gestellt werden sollen (siehe Abb. 6): (1) Automatische *Erkennung* von verdächtigen Mustern, (2) *Hervorheben* relevanter Nutzeroberflächenelemente, (3) *Erklärungen* über das beobachtete Muster und warum dieses unerwünscht ist sowie (4) *Vorschläge* von möglichen nächsten Schritten. Unsere Implementierung von GE im AntTracks Analyzer Werkzeug bietet diese Operationen auf jedem Schritt durch die Speicherleckanalyse sowie durch die Speicherfluktationsanalyse, um so Nutzer bestmöglich durch den kompletten Analyseprozess zu leiten.

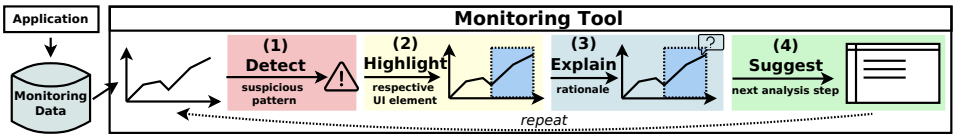


Abb. 6: Die vier Schritte der *Geleiteten Exploration*: (1) Erkennung, (2) Hervorheben, (3) Erklärungen, und (4) Vorschläge.

4 Zusammengefasste Kontributionen

Diese Thesis präsentiert (1) Algorithmen und Datenstrukturen zum Nutzbarmachen und Aggregieren von Speichertraces, (2) einen semi-automatischen Ansatz zum Erkennen und Inspizieren von verdächtig wachsenden Datenstrukturen, (3) verschiedene Visualisierungstechniken (basierend auf Zeitseriendiagrammen, 2D Baumvisualisierungen und der 3D Softwarestadt-Metapher), um die Speicherentwicklung einer Applikation zugänglicher und greifbarer zu machen, (4) eine Technik zum Erkennen und Inspizieren von hoher Speicherfluktation, (5) eine Nutzerstudie und strukturierte kognitive Durchgänge zur Evaluierung des Nutzens unserer präsentierten Techniken, sowie generelle Empfehlungen für Entwickler

von Speicherwerkzeugen, welche final zu (6) unserem Konzept der Nutzerunterstützung *Geleitete Exploration* führten, das speziell unerfahrene Benutzer durch Speicheranalysen leiten soll.

Ein Ziel dieser Arbeit war hervorzuheben, wie flexibel Speicheraufzeichnungen sein können. Wir zeigen interessante Anwendungsfälle, in denen diese eingesetzt werden, um Entwickler dabei zu unterstützen, Speicheranomalien aufzuspüren, zu untersuchen und zu beheben. Darüber hinaus zeigen wir, dass zeitliche Information, d.h., Information über die Entwicklung des Heaps über Zeit, von äußerstem Nutzen sein kann, um detaillierte Speicheranalysen wie beispielsweise Datenstrukturwachstums- und trendanalysen durchzuführen. Solche Analysen über Zeit sind nur mit Speichertraces möglich, da herkömmliche Speicherauszüge (heap dumps) es nicht erlauben, die Entwicklung von Objekten über einen Zeitraum zu beobachten. Auch wenn die Erzeugung von Speichertraces derzeit noch nicht weit verbreitet ist, so hoffen wir, dass Forschung wie diese als Motivation dienen kann, diesen Umstand zu ändern.

Neben theoretischen Konzepten resultierte die Thesis in einer Vielzahl von technischen Kontributionen: die Verbesserung der *AntTracks VM*, die Entwicklung des *AntTracks Analyzer* Werkzeugs, unserem 3D-Visualisierungswerkzeug *Memory Cities*[3], sowie unserem Web-basiertem 2D-Visualisierungswerkzeug *WebTreeViz*[4], welche alle öffentlich verfügbar sind. Wir nutzten diese Werkzeuge als Machbarkeitsnachweise für alle präsentierten Ideen, um deren Anwendbarkeit und Nutzen zu unterstreichen.

Literatur

[BG03] Blackwell, A.; Green, T.: Notational Systems — The Cognitive Dimensions of Notations Framework. In: HCI Models, Theories, and Frameworks. Interactive Technologies, 2003.

[Du03] Dufour, B.; Driesen, K.; Hendren, L.; Verbrugge, C.: Dynamic Metrics for Java. In: OOPSLA. 2003.

[He06] Hertz, M.; Blackburn, S. M.; Moss, J. E. B.; McKinley, K. S.; Stefanović, D.: Generating Object Lifetime Traces with Merlin. ACM Trans. Program. Lang. Syst. 28/3, Mai 2006.

[LBM15] Lengauer, P.; Bitto, V.; Mössenböck, H.: Accurate and Efficient Object Tracing for Java Applications. In: ICPE. 2015.

[LBM16] Lengauer, P.; Bitto, V.; Mössenböck, H.: Efficient and Viable Handling of Large Object Traces. In: ICPE. 2016.

[Le16] Lengauer, P.; Bitto, V.; Fitzek, S.; Weninger, M.; Mössenböck, H.: Efficient Memory Traces with Full Pointer Information. In: PPPJ. 2016.

[3] Video: `http://ssw.jku.at/General/Staff/Weninger/AntTracks/VISSOFT20/MemoryCities.mp4`

[4] Werkzeug:`http://bit.ly/STAG-MemoryTreeVizTool`; Video: `http://bit.ly/STAG-MemoryTreeVizVideo`

[Li19] Limberger, D.; Scheibel, W.; Döllner, J.; Trapp, M.: Advanced Visual Metaphors and Techniques for Software Maps. In: VINCI. 2019.

[MBR10] Maxwell, E. K.; Back, G.; Ramakrishnan, N.: Diagnosing Memory Leaks using Graph Mining on Heap Dumps. In: SIGKDD. 2010.

[Or21] Oracle: The HotSpot Group, last visited on 2022-02-03, 2021.

[PH16] Peiris, M.; Hill, J. H.: Automatically Detecting Excessive Dynamic Memory Allocations Software Performance Anti-Pattern. In: ICPE. 2016.

[RGM13] Ricci, N. P.; Guyer, S. Z.; Moss, J. E. B.: Elephant Tracks: Portable Production of Complete and Precise GG Traces. In: ISMM. 2013.

[Sc11] Schulz, H.-J.: Treevis.net: A Tree Visualization Reference. IEEE Computer Graphics and Applications 31/6, S. 11–15, 2011.

[SW00] Smith, C. U.; Williams, L. G.: Software Performance Antipatterns. In: WOSP. 2000.

[We21] Weninger, M.: Erkennung und Analyse von Speicheranomalien in Sprachen mit automatischer Speicherverwaltung unter Nutzung von Trace-basierter Speicherüberwachung, Dissertation, Institut für Systemsoftware / Johannes Kepler Universität Linz, 2021.

[WL07] Wettel, R.; Lanza, M.: Visualizing Software Systems as Cities. In: VISSOFT. S. 92–99, 2007.

[XR13] Xu, G.; Rountev, A.: Precise Memory Leak Detection for Java Software Using Container Profiling. ACM Trans. Softw. Eng. Methodol. 22/3, 2013.

[Xu13] Xu, G.: Resurrector: A Tunable Object Lifetime Profiling Technique for Optimizing Real-world Programs. In: OOPSLA. 2013.

Markus Weninger wurde am 4. April 1992 geboren. Seine Informatikausbildung begann er an der Höheren Technischen Bundeslehranstalt Leonding für *EDV & Organisation*. Danach absolvierte er das Bachelorstudium *Informatik* und das Masterstudium *Computer Science* mit Schwerpunkt *Software Engineering* an der Johannes Kepler Universität (JKU) Linz. Während seines Masterstudiums sammelte er bereits erste Forschungserfahrung am *Institut für Systemsoftware*, wo er sich mit dem Aufbereiten von Speicherüberwachungsdaten beschäftigte. Aus dieser Forschung ging anschließend seine Dissertation zum Thema Trace-basierte Speicheranalyse hervor, welche er im Juli 2021 mit ausgezeichnetem Erfolg verteidigte. Weiters wurde seine Dissertation mit dem *Award of Excellence* des österreichischen Bundesministerium für Bildung, Wissenschaft und Forschung ausgezeichnet. Derzeit arbeitet Markus als Senior Lecturer im Bereich Computer Science an der JKU und vereint dort im Unterrichten von Fächern wie *Softwareentwicklung*, *Compilerbau* oder *Algorithmen und Datenstrukturen* seine Liebe zur Informatik und zur Lehre.

Erkennung von Anomalien und Veränderung in Graphsequenzen[1]

Daniele Zambon[2]

Abstract:

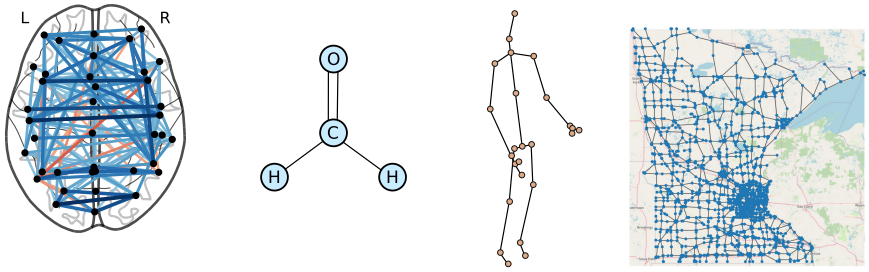
Wir verzeichnen einen erheblichen Zuwachs an Daten, die von Sensornetzen und sozialen Netzwerken gesammelt werden, verursacht durch technologische Entwicklungen und die Verbreitung sozialer Plattformen. Die Auswertung dieser riesigen Datenströme ist eine wichtige Aufgabe für die Wissenschaft ebenso wie für die Industrie. Da Datenströme von Sensoren (sei es physischen oder virtuellen) in der Regel funktionale Abhängigkeiten aufweisen, erweisen sich Graphen als reichhaltige Strukturen, die in der Lage sind, sowohl Informationen auf der Ebene der Sensoren/Entitäten als auch die komplexen Beziehungen zwischen den Entitäten zu modellieren. Diese graphbasierte Repräsentation wiederum ermöglicht uns, mittels Graph Neural Networks und Geometric Deep Learning, Inferenzen in Bezug auf Graphsequenzen anzustellen. Im Allgemeinen gehen solche Verarbeitungsverfahren allerdings von der Hypothese der Stationarität aus, die nicht immer gegeben ist, z.B. wenn eine Alterung der Sensoren, eine zeitliche Varianz oder eine Veränderungen in den Präferenzen der Nutzer auf sozialen Plattformen vorliegt. In dieser Dissertation befassen wir uns mit dem Problem der Identifizierung von Veränderungen der Stationarität, die durch unbekannte Phänomene im zugrundeliegenden Datenerzeugungsprozess verursacht werden und sich in der Sequenz der Graphen zeigen. Die wissenschaftlichen Ergebnisse erlauben es uns, auch das Problem der Erkennung von Anomalien zu behandeln, das in der Tat eine wertvolle Fortsetzung der Forschung darstellt. Wir betrachten eine allgemeine Familie von mit Attributen versehenen Graphen mit nicht-identifizierten Knoten, um ein möglichst breites Spektrum von Anwendungen abzudecken. Der Hauptbeitrag dieser Arbeit besteht in einer Methodik zur Verarbeitung einer Sequenz von Graphen, um unerwartete Ereignisse (Änderungen der Stationaritäten und/oder Auftreten von Anomalien) im Datenerzeugungsprozess zu erkennen. Die Methodik beruht auf der Entwicklung neuartiger Embeddings auf Graphenebene und Methoden zur Erkennung von Veränderungen, die durch ein solides theoretischen Grundgerüst unterstützt werden.

1 Einleitung

Immer mehr Daten stehen zur Verfügung, die von Monitoringinstanzen und Einzelpersonen gesammelt werden; die Daten stammen aus dem Internet der Dinge, den Neurowissenschaften und der Teilchenphysik, sowie aus Empfehlungssystemen und sozialen Netzwerken [AT05; BS16; JM15; Ra18], um nur einige datenerzeugende Anwendungen zu nennen.

[1] Englischer Titel der Dissertation: “Anomaly and Change Detection in Sequences of Graphs”

[2] Università della Svizzera italiana, Faculty of Informatics, Via Buffi 13, 6900 Lugano, Switzerland daniele.zambon@usi.ch



(a) (b) (c) (d)

Abb. 1: Beispiele für graphbasierte Repräsentationen. Tafel 1a) Mit der Pearson-Korrelation berechnetes Netzwerk funktioneller Konnektivität. Tafel 1b) Molekulargraph der Formaldehydverbindung. Tafel 1c) Skelettgraph eines Menschen. Tafel 1d) Graph der Hauptstraßen im Bundesstaat Minnesota.

Da Datenströme häufig zeitliche und funktionale Abhängigkeiten aufweisen, widmet die Forschung immer mehr Aufmerksamkeit den Inferenzmethoden, die bestehende Beziehungen ausnutzen. Innerhalb dieser Forschungslinie werden Daten effektiv als Graphen dargestellt, wie es natürlicherweise in Molekülen und Proteinen der Fall ist, wo Atome (oder Substrukturen) durch Bindungen vernetzt sind. [Bo05; Li16; Yo18], oder in Sensornetzwerken, wo erfasste Signale eine funktionale Abhängigkeit aufzeigen können [ANR13]. Andere Beispiele, bei denen eine Graphdarstellung nahe liegt, sind soziale Netzwerke, Smart Grids und körperbasierte Netzwerke [BS16; CML11; MH19; Po16; YXL18]. Graphdarstellungen ermöglichen ein relationales induktives Bias [Ba18; Mi80], das den Werkzeugen maschinellen Lernens ermöglicht, Vorwissen und bestehende Einschränkungen direkt während der Lernphase zu nutzen, indem sie z.B. das Vorhandensein einer Netzwerkstruktur ausnutzen [Li18].

In der Graphenmodellierung können funktionale Abhängigkeiten als ein einfaches skalares Gewicht beschrieben werden, das jeder Kante zugeordnet ist, die funktional verwandte Knoten verbindet, was zu Graphen mit Knoten und/oder Kanten führt, die mit Attributen versehen sind. Allgemeiner ausgedrückt, sind Attribute Kennzeichnungen oder Merkmale in Form von Skalaren, Vektoren, Klassenmitgliedschaften und benutzerdefinierten Datenstrukturen. Darüber hinaus können mehrere Attribute mit demselben Knoten oder derselben Kante assoziiert sein. Die Definition von mit Attributen versehenen Graphen deckt eine breite Familie von Graphen ab, zu der u. a. gerichtete und ungerichtete gelabelte Graphen mit einer variable Anzahl von Knoten und Kanten gehören. In Abbildung 1 sind einige Beispiele für mit Attributen versehenen Graphen dargestellt, bei denen die Graphenstruktur a priori bekannt ist oder direkt aus den Daten geschätzt wird, e.g. aus Pearson-Korrelationen, die in einer multivariaten Zeitreihe gemessen wurden.

In Szenarien, die lebenslange Datenströme erzeugen, wie die von cyber-physischen Systemen, kann die überwachte Umgebung in der Tat zeitlichen Verschiebungen in der Datenverteilung unterliegen, die z.B. durch saisonale Schwankungen, Alterung von Sensoren und zeitliche

Varianz der Interaktion zwischen der Umgebung und dem Datenerfassungssystem verursacht werden. Die Hypothese der Stationarität des datenerzeugenden Prozesses, die oft bei vielen Aufgaben des maschinellen Lernens angenommen wird, ist nicht unbedingt allgemein gültig, und wir müssen entweder ihre Gültigkeit im Verlauf der Zeit überwachen oder mit Mitigationsstrategien in die Machine-Learning Lösung eingreifen, um den veränderten Betriebsbedingungen Rechnung zu tragen. Der letztgenannte Fall wird als Lernen in nicht-stationären Umgebungen bezeichnet, und Anpassungsmechanismen sind einfache Beispiele für Mitigationsstrategien [Di15].

Beim Lernen in nicht-stationären Umgebungen unterscheiden wir zwischen passiven adaptiven Methoden, die sich kontinuierlich anpassen, sobald neue Daten beobachtet werden [EP11; LP17], und pro-aktiven Methoden, welche die Machine-Learning Modelle nur dann neu konfigurieren, wenn eine Änderung der Stationarität erkannt wird [ABR13; BG07]. Die Erkennung von *Änderungen der Stationarität* des datenerzeugenden Prozesses ermöglicht es uns zu ermitteln, wann das aktuelle Modell nicht mehr aktuell ist und seine (Hyper-)Parameter entweder aktualisiert oder sogar das gesamte Modell neu konfiguriert werden muss. Veränderungen der Stationarität treten in verschiedenen Formen auf. Eine abrupte Änderung der Stationarität bezeichnet beispielsweise einen Wechsel von einem stationären Zustand zu einem neuen - anderen - stationären Zustand [BN+93]. In anderen Fällen handelt es sich um ein vorübergehendes Verhalten, bei dem der Prozess von einem stationären Zustand zu einem anderen wechselt, bevor er wieder in seinen ursprünglichen Zustand zurückkehrt. Drifttypen von Veränderungen der Stationarität beziehen sich stattdessen auf langsame Übergänge des datenerzeugenden Prozesses [Ga14]. Ein Problem im Zusammenhang mit der Erkennung vorübergehender Veränderungen ist die Erkennung von Anomalien, die sich auf die Identifizierung von Transienten bezieht, die durch (fast) augenblickliche Ereignisse gekennzeichnet sind [Pi14]. Darüber hinaus kann es bei Graphen zu Ereignissen kommen, die Knoten, Kanten oder ganze Teilgraphen betreffen. Andere Arten von Veränderungen haben einen globalen Einfluss oder können nicht auf bestimmte Teile des Graphen zurückgeführt werden [ATK15; Ra15].

Methoden zur Erkennung von Anomalien und Veränderungen in Graphdaten ermöglichen nicht nur eine Modellanpassung, sondern bieten auch spezielle Werkzeuge zur Untersuchung physikalischer Phänomene, z.B. zur Überprüfung des baulichen Zustands eines Gebäudes [SCF00], zur Identifizierung von Fehlern in großen cyber-physischen Systemen [ANR16] oder zur Analyse der elektrische Aktivität im Gehirn, um über internes Funktionsverhalten Aufschluss zu geben [Ri13].

2 Forschungsproblem

In dieser Dissertation [Za22] betrachten wir einen zeitdiskreten stochastischen Prozess $\mathcal{P}$, der einen Graphen g_t im Zeitschritt t, $t \in \mathbb{N}$ erzeugt. Wir nehmen an, dass die erzeugte Sequenz $(g_t)_{t\in\mathbb{N}} = g_1, g_2, \ldots, g_t, \ldots$ aus einer stationären, normalerweise unbekannten, Verteilung P_0 gezogen wird. Wir befassen uns mit dem Problem der Identifizierung

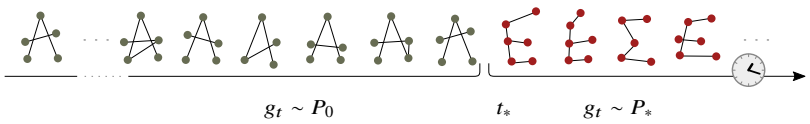


Abb. 2: Ein Beispiel für eine Graphsequenz, die im Zeitschritt t_* eine Änderung der Stationarität erfährt. Die Graphen g_t für $t < t_*$ werden unabhängig und identisch aus einer nominalen Verteilung P_0 gezogen und erzeugen Graphen, die dem Buchstaben "A" ähneln. Graphen g_t für $t \geq t_*$, die nach dem Änderungspunkt erzeugt werden, werden aus einer anderen Verteilung $P_* \neq P_0$ gezogen und ähneln dem Buchstaben "E".

möglicher Änderungen in der Stationarität, die durch Phänomene verursacht werden, die im zugrunde liegenden stochastischen Prozess auftreten und sich in der Graphsequenz zeigen. Formalisieren lässt sich dieses Problem als Erkennung, ob es einen Zeitschritt $t_* \in \mathbb{N}$ gibt, bei dem

$$g_t \sim \begin{cases} P_0, & t < t_* \\ P_t, & t \geq t_*, \qquad P_{t_*} \neq P_0. \end{cases} \tag{1}$$

Mit anderen Worten, der Prozess $\mathcal{P}$ befindet sich in einem stationären Zustand, der durch die Verteilung P_0 bestimmt wird, bis zum Zeitschritt t_*, wenn eine neue Datenverteilung $P_{t_*} \neq P_0$ auftritt. Nach dem Änderungszeitpunkt t_* kann das Verhalten des Prozesses $\mathcal{P}$ beliebig sein. In der Tat erlaubt die Formulierung (1) die Modellierung verschiedener Arten von Veränderungen. Beispielsweise treten abrupte Änderungen auf, wenn für die Verteilungen nach der Änderung $P_t = P_*$ für $t \geq t_*$ gilt, also

$$g_t \sim \begin{cases} P_0, & t < t_* \\ P_*, & t \geq t_*, \qquad P_* \neq P_0, \end{cases} \tag{2}$$

wie in Abbildung 2 dargestellt. Ebenso können wir Anomalien modellieren, die als sofortige vorübergehende Ereignissen behandelt werden, für welche die Graphen gemäß P_0 verteilt sind, außer für den Zeitschritt t_*, für den gilt $g_{t_*} \sim P_*$:

$$g_t \sim \begin{cases} P_0, & t \neq t_* \\ P_*, & t = t_*, \qquad P_* \neq P_0. \end{cases} \tag{3}$$

Es sei darauf hingewiesen, dass kurzzeitige nichtstationäre Störungen, die eine kleine Anzahl von Graphen aus einer anderen Verteilung P_* erzeugen, ebenfalls als Anomalien behandelt werden können. Jedoch sollten Anomalien von Ausreißern unterschieden werden. Ausreißer sind seltene Beobachtungen, die aus der Nominalverteilung P_0 gezogen sind und mit einer geringen Likelihood assoziiert sind, während anomale Graphen aus einer von P_0 abweichenden Verteilung P_* gezogen sind [Pi14]. Wenn die Anzahl der Beobachtungen, die mit dem Rest der Daten nicht übereinstimmen, zu gering ist, dann ist die Unterscheidung zwischen Anomalien und Ausreißern schwierig, wenn nicht gar unmöglich, obwohl es sich um konzeptionell unterschiedliche Arten von Beobachtungen handelt und sie als solche betrachtet werden sollten.

Graphen treten heutzutage in unterschiedlichsten Formen auf, die sich aus der Vielzahl der realen Anwendungen ergeben. Wir tragen dieser Variabilität Rechnung, indem wir die Dissertation für das allgemeinene Szenario formulieren, in dem die Topologie von einem Zeitschritt zum anderen variieren kann und in dem sowohl Knoten als auch Kanten mit jeglicher Art von Attribut versehen sein können, einschließlich Vektoren und nicht-numerischen Labels. Darüber hinaus kann eine Eins-zu-eins-Entsprechung zwischen Knoten verschiedener Graphen fehlen oder nicht gegeben sein, was den Vergleich von Graphen zu einem nicht-trivialen und rechnerisch schwierigen Problem macht; ein solcher Mangel an Knoten-Entsprechung wird als Graphen mit nicht identifizierten Knoten bezeichnet.

Soweit wir wissen, gibt es in der Literatur keine Arbeit, die sich mit Problemen der Erkennung von Veränderungen und Anomalien befasst, indem sie generische Familien von Graphen betrachtet, wie wir es in dieser Forschung tun. Der beschriebene Aufbau erfordert die Entwicklung eines dedizierten theoretischen Rahmens und mathematischer Werkzeuge.

3 Herausforderungen

Die Verarbeitung von Sequenzen von Graphen mit generischen Attributen und fehlender Korrespondenz der Knoten im Zeitverlauf ist wesentlich komplizierter als der Umgang mit Vektordaten. Wir identifizieren vier Hauptherausforderungen, welche die Identifizierung von Veränderungen der Stationarität in solchen Sequenzen zu einem schwierigen Problem machen.

C1 Graphenraum: Die erste Herausforderung besteht darin, die Geometrie des Graphenraums zu verstehen, wenn Graphen mit Attributen versehen sind und nicht identifizierte Knoten haben. Solche Werkzeuge ermöglichen wiederum Methoden des maschinellen Lernens, die geeignete Repräsentationen von Graphen liefern können, wodurch die nachgelagerten Aufgaben in der Praxis leichter zu lösen sind.

C2 Hypothesentests: Entwurf von statistischen Tests, die auf Graphsequenzen anwendbar sind. Geeignete Graphenstatistiken werden benötigt, um zwischen verschiedenen Verteilungen zu unterscheiden. Solche Statistiken sollten in der Lage sein, Stichproben von Graphen (Mengen von Graphen) zu verarbeiten, aber auch vielseitig genug sein, um in Streaming-Szenarien, d.h. sequentiell, zu arbeiten.

Tests, die gegen jede Alternativhypothese konsistent sind, erfordern Graphenmaße wie Distanzen und Kernels, die zumindest aussagekräftig genug sind, um zwischen Paaren nicht-isomorpher Graphen zu unterscheiden; wenn die Tests über eine Embedding-Abbildung entworfen werden, muss die Abbildung injektiv sein. Diese Eigenschaften sind schwer zu erfüllen, da sie von der kombinatorischen Natur von Graphen herrühren. Daher haben wir zwei weitere Herausforderungen identifiziert.

C3 Rechnerische Komplexität: Entwicklung von rechnerisch machbaren Methoden und robuster Näherungsmethoden. Dies kann den Unterschied ausmachen zwischen einer wirksamen Methode, die zu komplex ist, um in der Praxis angewandt zu werden, und einer Methode, die mit gewissen Einschränkungen eingesetzt werden kann und reale Probleme löst.

C4 Statistische Aussagekraft: Entwicklung dateneffizienter Methoden und Verfahren zur Bewältigung von Szenarien, bei denen große Graphen beteiligt sind und eine kleine Anzahl von Graphen gegeben ist. Die Ausnutzung von Eigenschaften der gegebenen Graphen, wie Symmetrien und Zwangsbedingungen, und Resampling-Methoden können die Aussagekraft der Tests erheblich steigern.

Wir haben uns die Herausforderung **C2** als zentrales Ziel der Forschung gesetzt. Die Bewältigung von **C2** wird jedoch durch Fortschritte bei den Herausforderungen **C1**, **C3** und **C4** erleichtert. In der Tat bietet **C1** mathematische Werkzeuge, um Graphen in geeigneten Bereichen zu behandeln und ihre inhärenten Eigenschaften zu nutzen. Auch wenn theoretische Werkzeuge zur Verfügung stehen, sind sie nicht zwangsläufig konstruktiv berechenbar, ebenso mag es sein, dass sie keinen effektiven und effizienten Algorithmus für die Bewertung haben, oder sie auf großen Stichproben basieren; hier kommen **C3** und **C4** ins Spiel.

4 Thesisbeiträge

Die in dieser Dissertation [Za22] vorgestellten Arbeiten wurden wissenschaftlichen Konferenzen und Journalen veröffentlicht. Zusammengefasst sind die Beiträge der Dissertation die folgenden.

Methodik: Wir formalisieren eine Methodik zur Durchführung von Hypothesentests auf Graphsequenzen, wobei wir die Auswirkungen einer Verlagerung der Analyse von der Domäne der Graphen auf eine zugänglichere Domäne, wie einen Vektorraum oder eine Riemannsche Mannigfaltigkeit, untersuchen [ZAL18; ZAL19]. Die Methodik ermöglicht sowohl die Untersuchung der verfügbaren Methoden zur Erkennung von Veränderungen als auch die Entwicklung neuer Techniken.

Embeddings auf Graphebene: Wir geben einen Überblick über wichtige Beziehungen zwischen Graph-Distanzen, -Kernels und -Embeddings und konzentrieren uns bei der Analyse auf Embedding-Methoden, da diese die genannten Vorteile bieten. Wir untersuchen verschiedene Methoden der Graph-Embeddings, die sich in die obige Methodik einfügen [Gr19; ZAL20; ZLA18].

Tests zur Erkennung von Veränderungen: Wir zeigen, wie die Methodik durch die Kombination verschiedener Embedding-Methoden mit verschiedenen statistischen Tests in spezifische Tests umgewandelt werden kann [Gr19; ZAL18; ZAL19; ZLA18]. In diesem Zusammenhang schlagen wir Tests zur Erkennung von Änderungen des Mittelwerts der Graphenverteilung sowie von beliebigen Änderungen der Verteilung vor. Wir schlagen statistische Tests vor, die auf Mannigfaltigkeiten operieren, sowie Ensemble-Tests und Tests zur Erkennung mehrerer Änderungen.

Die vorgeschlagenen Methoden wurden anhand verschiedener Arten von Graphsequenzen validiert, darunter Sequenzen synthetischer Graphen [ZLA17; ZLA18], Molekülen [ZAL18; ZAL20] und Netzwerken im Gehirn [Gr19; ZAL19].

5 Ausblick

Wir sind der Meinung, dass es wichtig ist, allgemeine und robuste Ansätze zu entwickeln, die den Benutzer von der Aneignung spezieller fachlicher Kenntnisse entlasten und somit eine möglichst breite Anwendbarkeit der Methoden ermöglichen. Es gibt Methoden zur Erkennung von Veränderungen, die gegenüber jeder Veränderung der Verteilung konsistent sind. Solche Methoden sind daher auf praktisch jede Problemstellung und jede Art von Graphen anwendbar, mit nachweislich erwartbarer Leistung. Wir sind uns jedoch auch bewusst, dass sie nicht unbedingt praktikabel sind, da sie möglicherweise eine unrealistische Menge an Daten erfordern oder einfach nicht genügend statistische Aussagekraft für das gegebene Problem haben, was zu langen Erkennungsverzögerungen führt, die den gegebenen Anwendungsanforderungen nicht entsprechen. Umgekehrt können Methoden, die auf spezifische Probleme zugeschnitten sind, in der Praxis per Definition bessere Ergebnisse erzielen als allgemeine Methoden, vorausgesetzt, man ist in der Lage, das verfügbare Wissen effektiv zu nutzen und die geeignete Methode für das jeweilige Problem auszuwählen.

Mit diesem Kommentar betonen wir, dass der methodische Beitrag dieser Doktorarbeit ein Grundgerüst für die Entwicklung neuer maßgeschneiderter Tests bietet und die vorgeschlagenen Methoden insgesamt gute Ausgangspunkte sind, die nach Bedarf weiter angepasst und erweitert werden können. Zukünftige Forschung sollte sich daher auf Techniken konzentrieren, mit denen (i) relevantes Vorwissen ausgewählt, (ii) in die Architektur der maschinellen Lernmodelle integriert und (iii) im Lernprozess wirksam ausgenutzt werden kann. Zu den besten Möglichkeiten, die ich mir von dieser Arbeit verspreche, gehört die Verwendung der vorgeschlagenen Graph Random Neural Features [ZAL20] als vielseitige Vorlage für die Entwicklung neuer Embedding-Methoden; zum Beispiel durch die Identifizierung einer grundlegenden Familie von Graphenfunktionen, die in der Lage sind, Symmetrien und Einschränkungen in der gegebenen spezifischen Problemstellung auszunutzen.

Literatur

[ABR13] Alippi, C.; Boracchi, G.; Roveri, M.: Just-in-time classifiers for recurrent concepts. IEEE transactions on neural networks and learning systems 24/4, S. 620–634, 2013.

[ANR13] Alippi, C.; Ntalampiras, S.; Roveri, M.: A cognitive fault diagnosis system for distributed sensor networks. IEEE transactions on neural networks and learning systems 24/8, S. 1213–1226, 2013.

[ANR16] Alippi, C.; Ntalampiras, S.; Roveri, M.: Model-free fault detection and isolation in large-scale cyber-physical systems. IEEE Transactions on Emerging Topics in Computational Intelligence 1/1, S. 61–71, 2016.

[AT05] Adomavicius, G.; Tuzhilin, A.: Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE transactions on knowledge and data engineering 17/6, S. 734–749, 2005.

[ATK15] Akoglu, L.; Tong, H.; Koutra, D.: Graph based anomaly detection and description: a survey. Data mining and knowledge discovery 29/3, S. 626–688, 2015.

[Ba18] Battaglia, P. W.; Hamrick, J. B.; Bapst, V.; Sanchez-Gonzalez, A.; Zambaldi, V.; Malinowski, M.; Tacchetti, A.; Raposo, D.; Santoro, A.; Faulkner, R. et al.: Relational inductive biases, deep learning, and graph networks. arXiv preprint arXiv:1806.01261/, 2018.

[BG07] Bifet, A.; Gavalda, R.: Learning from time-changing data with adaptive windowing. In: Proceedings of the 2007 SIAM international conference on data mining. SIAM, S. 443–448, 2007.

[BN+93] Basseville, M.; Nikiforov, I. V. et al.: Detection of abrupt changes: theory and application. prentice Hall Englewood Cliffs, 1993.

[Bo05] Borgwardt, K. M.; Ong, C. S.; Schönauer, S.; Vishwanathan, S.; Smola, A. J.; Kriegel, H.-P.: Protein function prediction via graph kernels. Bioinformatics 21/suppl 1, S. i47–i56, 2005.

[BS16] Bastos, A. M.; Schoffelen, J.-M.: A tutorial review of functional connectivity analysis methods and their interpretational pitfalls. Frontiers in systems neuroscience 9/, S. 175, 2016.

[CML11] Cho, E.; Myers, S. A.; Leskovec, J.: Friendship and mobility: user movement in location-based social networks. In: Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. S. 1082–1090, 2011.

[Di15] Ditzler, G.; Roveri, M.; Alippi, C.; Polikar, R.: Learning in nonstationary environments: a survey. IEEE Computational Intelligence Magazine 10/4, S. 12–25, 2015.

[EP11] Elwell, R.; Polikar, R.: Incremental learning of concept drift in nonstationary environments. IEEE Transactions on Neural Networks 22/10, S. 1517–1531, 2011.

[Ga14] Gama, J.; Žliobaitė, I.; Bifet, A.; Pechenizkiy, M.; Bouchachia, A.: A survey on concept drift adaptation. ACM computing surveys (CSUR) 46/4, S. 1–37, 2014.

[Gr19] Grattarola, D.; Zambon, D.; Alippi, C.; Livi, L.: Change Detection in Graph Streams by Learning Graph Embeddings on Constant-Curvature Manifolds. IEEE Transactions on Neural Networks and Learning Systems/, 2019.

[JM15] Jordan, M. I.; Mitchell, T. M.: Machine learning: Trends, perspectives, and prospects. Science 349/6245, S. 255–260, 2015.

[Li16] Livi, L.; Maiorino, E.; Giuliani, A.; Rizzi, A.; Sadeghian, A.: A generative model for protein contact networks. Journal of Biomolecular Structure and Dynamics 34/7, S. 1441–1454, 2016.

[Li18] Li, Y.; Yu, R.; Shahabi, C.; Liu, Y.: Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting. In: International Conference on Learning Representations. 2018, URL: https://openreview.net/forum?id=SJiHXGWAZ.

[LP17] Li, K.; Principe, J. C.: Transfer learning in adaptive filters: The nearest instance centroid-estimation kernel least-mean-square algorithm. IEEE Transactions on Signal Processing 65/24, S. 6520–6535, 2017.

[MH19] Masuda, N.; Holme, P.: Detecting sequences of system states in temporal networks. Scientific reports 9/1, S. 795, 2019.

[Mi80] Mitchell, T. M.: The need for biases in learning generalizations. Department of Computer Science, Laboratory for Computer Science Research . . ., 1980.

[Pi14] Pimentel, M. A.; Clifton, D. A.; Clifton, L.; Tarassenko, L.: A review of novelty detection. Signal Processing 99/, S. 215–249, 2014.

[Po16] Possemato, F.; Paschero, M.; Livi, L.; Sadeghian, A.; Rizzi, A.: On the impact of topological properties of smart grids in power losses optimization problems. International Journal of Electrical Power and Energy Systems 78/, S. 755–764, 2016.

[Ra15] Ranshous, S.; Shen, S.; Koutra, D.; Harenberg, S.; Faloutsos, C.; Samatova, N. F.: Anomaly detection in dynamic networks: A survey. Wiley Interdisciplinary Reviews: Computational Statistics 7/3, S. 223–247, 2015.

[Ra18] Radovic, A.; Williams, M.; Rousseau, D.; Kagan, M.; Bonacorsi, D.; Himmel, A.; Aurisano, A.; Terao, K.; Wongjirad, T.: Machine learning at the energy and intensity frontiers of particle physics. Nature 560/7716, S. 41–48, 2018.

[Ri13] Richiardi, J.; Achard, S.; Bunke, H.; Van De Ville, D.: Machine learning with brain graphs: predictive modeling approaches for functional imaging in systems neuroscience. IEEE Signal processing magazine 30/3, S. 58–70, 2013.

[SCF00] Sohn, H.; Czarnecki, J. A.; Farrar, C. R.: Structural health monitoring using statistical process control. Journal of structural engineering 126/11, S. 1356–1363, 2000.

[Yo18] You, J.; Liu, B.; Ying, Z.; Pande, V.; Leskovec, J.: Graph convolutional policy network for goal-directed molecular graph generation. In: Advances in neural information processing systems. S. 6410–6421, 2018.

[YXL18] Yan, S.; Xiong, Y.; Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: Thirty-second AAAI conference on artificial intelligence. 2018.

[Za22] Zambon, D.: Anomaly and Change Detection in Sequences of Graphs, Diss., Università della Svizzera italiana, 2022.

[ZAL18] Zambon, D.; Alippi, C.; Livi, L.: Concept Drift and Anomaly Detection in Graph Streams. IEEE Transactions on Neural Networks and Learning Systems/, S. 1–14, 2018.

[ZAL19] Zambon, D.; Alippi, C.; Livi, L.: Change-Point Methods on a Sequence of Graphs. IEEE Transactions on Signal Processing/, 2019.

[ZAL20] Zambon, D.; Alippi, C.; Livi, L.: Graph Random Neural Features for Distance-Preserving Graph Representations. In (III, H. D.; Singh, A., Hrsg.): Proceedings of the 37th International Conference on Machine Learning (ICML). Bd. 119. Proceedings of Machine Learning Research, PMLR, Virtual, S. 10968–10977, 2020, URL: http://proceedings.mlr.press/v119/zambon20a.html.

[ZLA17] Zambon, D.; Livi, L.; Alippi, C.: Detecting Changes in Sequences of Attributed Graphs. In: IEEE Symposium Series on Computational Intelligence. 2017.

[ZLA18] Zambon, D.; Livi, L.; Alippi, C.: Anomaly and Change Detection in Graph Streams through Constant-Curvature Manifold Embeddings. In: IEEE International Joint Conference on Neural Networks. 2018.

Daniele Zambon ist derzeit Post-Doktorand am Dalle Molle Institut für Künstliche Intelligenz (IDSIA, Schweiz). Seine Forschungsinteressen umfassen das Lernen von Graphenrepräsentationen, Graph Stream Mining und statistische Tests zur Erkennung von Anomalien und Veränderungen. Er promovierte in Informatik an der Università della Svizzera italiana (Schweiz) und erwarb den Master- und Bachelor-Abschluss an der Università degli Studi di Milano (Italien). Er war Gastforscher an der University of Florida (USA) und der University of Exeter (UK). Er war Praktikant bei STMicroelectronics (Italien), wo er seine Masterarbeit schrieb. Er war Mitglied des Programmkomitees hochrangiger Konferenzen und Fachzeitschriften, darunter IEEE TNNLS, IEEE TSP, IEEE PAMI, NeurIPS, ICLR, ICML, CVPR.