Verbesserte Algorithmen und Bedingte Untere Schranken für Probleme in Formaler Verifikation und Reaktiver Synthese[1]

Veronika Loitzenbauer[2]

Abstract: Die formale Verifikation ist ein Ansatz um Fehler in Computerprogrammen und anderen Systemen automatisiert und systematisch aufzuspüren oder die Erfüllung von gewünschten Eigenschaften zu garantieren. Reaktive Synthese bezeichnet die Erzeugung von Systemen basierend auf einer formalen Spezifikation, so dass sich diese wie gewünscht nach auen hin verhalten. In der Dissertation betrachten wir algorithmische Probleme in formaler Verifikation und reaktiver Synthese aus theoretischer Perspektive. Wir zeigen einerseits Algorithmen mit verbesserten asymptotischen Laufzeitschranken und anderseits bedingte untere Schranken, die zeigen, dass weitere Verbessungen der Laufzeitschranken zu Durchbrüchen im Algorithmendesign für wohlbekannte Probleme führen würden. Wir verbinden damit formale Methoden für Systeme mit Algorithmentheorie und führen neue Techniken und Forschungsrichtungen für Polynomialzeitprobleme in diesem Gebiet ein.

1 Einführung

Fehler im Software und Hardware Design können unsere Sicherheit gefährden und hohe Kosten verursachen wenn sie zum Beispiel bei der Steuerung eines Flugzeuges oder im Design eines Prozessors auftreten. Beweisbar fehlerfreie Systeme wären daher sehr wünschenswert. Wir betrachten hier abstrakte Modelle von Systemen. In der Praxis kann so ein System vieles sein, von einer einfachen Verkehrsampel bis zu parallelen Computerprogrammen, Kommunikationsprozessen oder elektronischen Schaltungen. Die Korrektheit von Systemen zu beweisen ist im Allgemeinen unmöglich, jedoch wurden in den letzten Jahrzehnten viele nützliche formale Methoden sowie Tools für die praktische Anwendung entwickelt, die helfen, die Korrektheit von Systemen zu überprüfen. Die *formale Verifikation* ist eine essentielle Komponente im iterativen Design von Systemen wie Mikroprozessoren, Kommunikationsprotokollen und sicherheitsrelevanten Algorithmen geworden. Die *Modellprüfung* ist ein vollautomatisierter Ansatz in der Verifikation um entweder die Korrektheit eines Systems bezüglich bestimmter Eigenschaften zu beweisen oder ein Gegenbeispiel dafür zu finden. Während Verifikation hauptsächlich für iterative Designprozesse verwendet wird, verlangt das *Syntheseproblem* die automatische Generierung von korrekten Systemen aus einer Spezifikation. Die *reaktive Synthese* ist die Synthese von Systemen, die wiederholt mit ihrer Umgebung interagieren und daher *reaktive Systeme* genannt werden. In dieser Arbeit beschäftigen wir uns mit algorithmischen Fragen im Bereich der Modellprüfung und Synthese.

[1] Englischer Titel der Dissertation: "Improved Algorithms and Conditional Lower Bounds for Problems in Formal Verification and Reactive Synthesis" [Lo16]. Siehe [Lo16] für detailierte Literaturverweise. Die Dissertation basiert auf den Publikationen [Ch14b, CHL17, Ch16b, Ch16a, Ch17].

[2] Institut für formale Modelle und Verifikation, Johannes Kepler Universität Linz, veronika.loitzenbauer@jku.at

Modelle. Für Verifikation und Synthese werden mathematische *Modelle* der Systeme sowie ein formale Beschreibung ihrer *Spezifikation*, das heißt ihres gewünschten Verhaltens, benötigt. Ein Modell eines Systems beschreibt typischerweise den Kontrollfluss sowie die Interaktion zwischen verschiedenen Teilen des Systems beziehungsweise zwischen dem System und seiner Umgebung, während Details der Implementierung ignoriert werden.

Endliche gerichtete Graphen sind ein Modell für nicht-deterministische Systeme. Die Knoten eines Graphen repräsentieren dabei die Zustände des Systems und die Kanten die bergänge zwischen den Zuständen. Die verschiedenen ausgehenden Kanten eines Knotens repräsentieren verschiedene Verhaltensmöglichkeiten des Systems die, zum Beispiel, durch die Nicht-Determiniertheit in parallelen oder verteilten Ausführungen von Prozessen oder, während dem iterativen Design von Systemen, durch unterschiedliche Designmöglichkeiten entstehen können.

Markow-Entscheidungsprozesse (*MEPs*) modellieren Systeme, die sowohl Nicht-Determiniertheit als auch zufälliges Verhalten zeigen. MEPs sind Graphen in denen eine Teilmenge der Knoten eine Wahrscheinlichkeitsverteilung über ihre ausgehenden Kanten besitzt. So eine Wahrscheinlichkeitsverteilung kann z.B. Randomisierung in verteilten Systemen, randomisierte Kommunikationsprotokolle, verlustbehaftete Kommunikationskanäle oder experimentelle Daten über die Umgebung des Systems modellieren.

In *Spielgraphen* sind die Knoten zwischen *zwei Spielerinnen* aufgeteilt. Typischerweise repräsentiert eine Spielerin das System und die anderen die Umgebung oder, wie in der Verifikation von Verzweigunszeit Eigenschaften von reaktiven Systemen, eine Spielerin modelliert die existenziellen Quantoren und eine die universellen Quantoren. Zwei-Spieler Spiele auf Graphen sind für viele Probleme in der Verifikation und Synthese wichtig, sowie der Synthese von reaktiven Systemen und der Verifikation von offenen Systemen.

Spezifikationen. Der *Automaten-basierte* Ansatz für Modelprüfung und Synthese ist eine anerkannte Methode um das erwünschte Verhalten von Systemen mit Hilfe von ω-*regulären Zielvorgaben* formal zu beschreiben (ω-reguläre Sprachen erweitern reguläre Sprachen auf unendliche Wörter) und kann die meisten häufig verwendeten Eigenschaften in formaler Verifikation und reaktiver Synthese ausdrücken. Jede Zielvorgabe hat eine komplementäre oder *duale* Zielvorgabe. Wenn eine Zielvorgabe das gewünschte Verhalten beschreibt, so beschreibt ihre duale Zielvorgabe das ungewünschte Verhalten. In Spielgraphen haben die beiden Spielerinnen zueinander komplementäre Zielvorgaben.

Die grundlegendsten Eigenschaften eines sicherheitskritischen Systems werden durch *Sicherheitszielvorgaben* beschrieben. Für Sicherheitszielvorgaben wollen wir verifizieren, dass eine gegebene Menge von schlechten Systemzuständen vermieden werden kann. Ein Beispiel für so einen schlechten Systemzustand in einem parallelen System wäre wenn zwei Prozesse gleichzeitig in einem kritischen (d.h. sequentiell zu absolvierenden) Abschnitt des Systems wären. Die duale Zielvorgabe zu einer Sicherheitszielvorgabe ist eine *Erreichbarkeitszielvorgage*, die eine Menge von guten Systemzuständen angibt, die letztendlich erreicht werden soll.

Die Frage ob jede Anfrage eines Prozesses einen kritischen Abschnitt zu durchlaufen schließlich erlaubt wird ist ein Beispiel für eine weitere Art von Eigenschaften, die mit *Büchi*-Zielvorgaben beschrieben wird. Dabei muss eine Menge von guten Zuständen *unendlich oft* erreicht werden. Die dazu duale Zielvorgabe ist die *co-Büchi*-Zielvorgabe, bei der eine Menge von schlechten Zuständen nur endlich oft erreicht werden darf.

Streett-Zielvorgaben und ihre dualen *Rabin*-Zielvorgaben können alle ω-regulären Sprachen ausdrücken. Streett-Zielvorgaben entsprechen direkt *starken Fairnessbedingungen*. Ein Scheduler, zum Beispiel, ist *stark fair* wenn jedes Event, das unendlich oft aufgerufen wird, auch unendlich oft eingeplant wird. Eine Streett-Zielvorgabe wird durch k Paare $(L_i, U_i)_{1 \leq i \leq k}$ von Knotenmengen definiert. Zur Erfüllung der Streett-Bedingung muss für alle k Paare ein unendlicher Pfad, der die Menge L_i unendlich oft kreuzt auch die Menge U_i unendlich oft kreuzen. Büchi- und co-Büchi- Zielvorgaben sind wichtige Spezialfälle von Streett- und Rabin Zielvorgaben mit $k = 1$.

Paritäts-Zielvorgaben verallgemeineren Büchi-Zielvorgaben und sind ebenso Spezialfälle von sowohl Streett- und Rabin-Zielvorgaben. Umgekehrt können auch Streett- und Rabin-Zielvorgaben in Paritäts-Zielvorgaben umgewandelt werden, jedoch wächst dabei die Modellgröße exponentiell. Die duale Zielvorgabe zu einer Paritäts-Zielvorgabe ist wieder eine Paritäts-Zielvorgabe. Paritäts-Zielvorgaben sind besonders wichtig da sie äquivalent zum modalen μ-Calculus sind, einer der wichtigsten Logiken für die Modellprüfung. Für eine Paritäts-Zielvorgabe werden den Knoten natürliche Zahlen, Prioritäten genannt, zugeordnet. Ein unendlicher Pfad erfüllt eine *gerade* Paritätsbedingung wenn die höchste Priorität, die unendlich oft vorkommt gerade ist, und ansonsten die komplementäre *ungerade* Paritätsbedingung.

Eine andere Verallgemeinerung von Paritäts-Zielvorgaben, die über ω-reguläre Zielvorgaben hinausgeht, sind *Mittelwerts*-Zielvorgaben. Hierfür werden den Kanten des Modells Gewichte zugeordnet und die Zielvorgabe wird bezüglich des Durchschnittsgewichts einer Sequenz von Kanten definiert. Mit solchen *quantitativen Zielvorgaben* kann zum Beispiel der durchschnittlichen Resourcenverbrauch oder die durchschnittliche Verzögerung eines Systems modelliert werden.

Algorithmische Fragestellungen. Um über alle möglichen Ausführungen eines Systems argumentieren zu können, betrachten wir unendliche Pfade, die durch das Bewegen eines Markers entlang der Kanten des Modells induziert werden. Die Wahl einer ausgehenden Kante an einem (nicht-zufälligen) Knoten wird als *Strategie* bezeichnet.

In *Graphen* wollen wir für jeden Ausgangsknoten wissen ob es einen unendlichen Pfad gibt, der die Zielvorgabe erfüllt. Die Menge der Ausgangsknoten, für die das der Fall ist, ist die *Gewinnmenge*. Die algorithmische Fragestellung für Graphen ist die Gewinnmenge sowie zugehörige Strategien zu berechnen.

In *MEPs* gibt es zusätzlich Zufallsknoten und an den Zufallsknoten wird die ausgehende Kante anhand der gegebenen Wahrscheinlichkeitsverteilung ausgewählt. In MEPs wollen wir für jeden Ausgangsknoten wissen ob es eine Strategie für die Nicht-Zufallsknoten gibt, so dass die Zielvorgabe mit Wahrscheinlichkeit 1 erfüllt wird. Die Menge der Ausgangs-

knoten, für die das der Fall ist, heißt *quasi-sichere Gewinnmenge*. Die Anforderung die Zielvorgabe mit Wahrscheinlichkeit 1 zu erfüllen wird auch die *qualitative* Analyse von MEPs genannt und ist in der Analyse von randomisierten verteilten Algorithmen üblich.

In *Spielgraphen* werden die unendlichen Pfade durch die Spielzüge der beiden Spielerinnen induziert, wobei jeweils die Eigentümerin eines Knotens entscheidet über welche ausgehende Kante eines Knotens der Marker bewegt wird. Eine *Strategie* einer Spielerin ist eine Funktion, die für jeden Knoten der Spielerin beschreibt wie ein gegebener Pfad fortgesetzt wird. Eine *Gewinnstrategie* stellt sicher dass die Zielvorgabe der Spielerin gegen jede mögliche Strategie der Gegnerin erreicht wird. Die *Gewinnmenge* einer Spielerin ist die Menge aller Ausgangsknoten für die sie eine Gewinnstrategie besitzt. Die Gewinnmengen der zwei Spielerinnen partitionieren den Spielgraphen. Die algorithmische Fragestellung ist es, die Gewinnmengen und die zugehörigen Gewinnstrategien zu bestimmen.

Symbolische Modellprüfung. Eine fundamentale Schwierigkeit bei der Modellprüfung ist, dass die Anzahl der Systemzustände, und damit die Anzahl der Knoten des Modells, exponentiell in der Anzahl der Systemvariablen ist. Ein Ansatz um damit umzugehen sind *symbolische* Algorithmen, bei denen die Systemzustände und -übergänge nicht explizit konstruiert werden sondern stattdessen die Vorgänger und Nachfolgerzustände von Mengen von Zuständen bei Bedarf generiert werden. Eine Menge von Zuständen kann hierfür zum Beispiel mit einem binären Entscheidungsdiagramm (BDD) kodiert werden. Für Spielgraphen mit Paritäts-Zielvorgaben (Paritätsspielen) betrachten wir auch symbolische Algorithmen.

Bedingte untere Schranken. In dieser Arbeit beschäftigen wir uns hauptsächlich mit Problemen, für die Polynomialzeitalgorithmen bekannt sind, die einzige Ausnahme sind Paritätsspiele. In der Komplexitätstherie gelten Polynomialzeitalgorithmen als effizient. Für sehr große Graphen wie sie in der Modellprüfung auftreten, kann aber zum Beispiel der Unterschied zwischen einem Algorithmus, der in quadratischer Zeit läuft, und einem, der in linearer Zeit läuft, entscheidend für seine praktische Anwendbarkeit sein. Deshalb würden wir gerne auch für Polynomialzeitprobleme Algorithmen mit besseren asymptotischen Laufzeitschranken finden oder zeigen, dass solche Algorithmen nicht existieren können. Unbedingte super-lineare untere Schranken für Polynomialzeitprobleme sind sehr selten. Jedoch wurden in den letzten Jahren *bedingte untere Schranken* für viele verschiedene kombinatorische Probleme gezeigt. Das sind untere Schranken, die auf einer Annahme über die bestmögliche Laufzeitschranke (abgesehen von Termen niedrigerer Ordnung) für ein wohluntersuchtes Problem basieren. Die bedingten unteren Schranken in dieser Arbeit nehmen an dass (A1) es keinen kombinatorischen Algorithmus für die Multiplikation von zwei $n \times n$ booleschen Matrizen mit einer Laufzeit von $O(n^{3-\varepsilon})$ für ein beliebiges $\varepsilon > 0$ gibt oder (A2) es für alle $\varepsilon > 0$ ein k gibt so dass es keinen Algorithmus für das k-CNF-SAT Problem gibt der in Zeit $2^{(1-\varepsilon)\cdot n} \cdot \mathrm{poly}(m)$ läuft, wo n die Anzahl der Variablen und m die Anzahl der Klauseln in der k-CNF-SAT Formel ist. Kombinatorisch bezeichnet hier die Vermeidung von bestimmten theoretisch schnellen (aber inpraktikablen) Matrixmultiplikationsalgorithmen. Annahme (A2) ist bekannt als Strong Exponential Time Hypothesis (SETH). Diese beiden Annahmen sind bereits für viele bedingte untere Schranken verwendet worden, zum Beispiel im Bereich von dynamischen Graphalgorithmen,

kontext-freier Grammatik und der Verifizierung von Grapheigenschaften erster Ordnung. Bisher ist keine Beziehung zwischen den beiden Annahmen (A1) und (A2) bekannt.

Für einige grundlegende Modellprüfungsprobleme sind die bestbekannten oberen Laufzeitschranken von quadratischer oder kubischer Ordnung und es gibt keine super-linearen unteren Schranken. In dieser Arbeit präsentieren wir mehrere Algorithmen, die verbesserte Laufzeitschranken bieten, und etablieren die ersten (super-linearen) bedingten unteren Schranken für fundamentale Polynomialzeitprobleme in der Modellprüfung und Synthese.

2 Forschungsstand

In diesem Abschnitt präsentieren wir einen berblick über die bereits bekannten algorithmischen Ergebnisse für die relevanten Modelle und Zielvorgaben. Wir bezeichnen mit n die Anzahl der Knoten und mit m die Anzahl der Kanten im gegebenen Modell. Für Paritäts-Zielvorgaben bezeichnen wir mit c die Anzahl der den Knoten zugeordneten Prioritäten und für Mittelwerts-Zielvorgaben bezeichnet W das maximale Gewicht einer Kante. Die hier angegebenen Laufzeitschranken berücksichtigen nicht die Abhängigkeit von anderen Eingabeparametern und sind zum Teil zu Gunsten der Lesbarkeit vereinfacht. Der detailierte Forschungsstand findet sich in den entsprechenden Kapiteln der Dissertation [Lo16].

Für *Graphen* kann die Gewinnmenge für Erreichbarkeits-, Sicherheits-, Büchi- und co-Büchi-Zielvorgaben in linearer Zeit ($O(m)$) berechnet werden, für Paritäts-Zielvorgaben in Zeit $O(m \log n)$ [CH14a]. Für Streett-Zielvorgaben gibt es einen $O(m \min(\sqrt{m \log n}, n))$ Zeit Algorithmus [HT96]. Der triviale Algorithmus für Rabin-Zielvorgaben benötigt Zeit $O(mn)$. Für Mittelwerts-Zielvorgaben laufen die besten Algorithmen in Zeit $O(mn)$ [Ka78] und $O(m\sqrt{n}\log(nW))$ [OA92].

Für *MEPs* sind Linearzeitalgorithmen nur für Sicherheits-Zielvorgaben bekannt, für welche das Problem äquivalent zu jenem in Spielgraphen ist. Die quasi-sichere Gewinnmenge für Erreichbarkeits-, Büchi- und co-Büchi-Zielvorgaben kann in Zeit $O(\min(m^{1.5}, n^2))$ berechnet werden [CJH03, CH14a] und mit einem zusätzlichen Faktor von $\log n$ auch für Paritäts-Zielvorgaben [CH14a]. Für Streett- und Rabin-Zielvorgaben folgen Laufzeiten von $O(n \cdot \min(m^{1.5}, n^2))$ aus [CH14a]. Mittelwerts-Zielvorgaben in MEPs können in polynomialer Zeit mit linearer Programmierung und in pseudo-polynomialer Zeit von $O(mnW)$ [FV97] gelöst werden.

In *Spielgraphen* ist die Laufzeit für eine Zielvorgabe und ihre duale Zielvorgabe jeweils die gleiche, da die beiden Zielvorgaben den Zielvorgaben der beiden Spielerinnen entsprechen und die beiden Gewinnmengen die Knoten partitionieren. Für Erreichbarkeits- und Sicherheitszielvorgaben können die Gewinnmengen in linearer Zeit berechnet werden. Für Büchi- und co-Büchi-Zielvorgaben benötigt der bestbekannte Algorithmus Zeit $O(n^2)$ [CH14a]. Für Streett- und Rabin-Zielvorgaben ist das Problem coNP- bzw. NP-vollständig [EJ88]. Für den Spezialfall von 1-Paar Streett und 1-Paar Rabin Zielvorgaben gibt es einen $O(mn)$ Zeit Algorithmus [Ju00]. Paritätsspiele und ihre Verallgemeinerung Mittelwertsspiele sind eine der wenigen "natürlichen" Probleme die in $\mathrm{NP} \cap \mathrm{coNP}$ liegen für die keine Polynomialzeitalgorithmen bekannt sind. Bis vor kurzem waren die

besten bekannten Algorithmen für Paritätsspiele ein $n^{O(\sqrt{n})}$-Zeit [JPZ08] und ein (ca.) $O(m \cdot n^{c/3})$-Zeit [Sc07] Algorithmus, nach Abschluss der Dissertation wurde der erste quasi-polynomial Zeit Algorithmus veröffentlicht [Ca17]. Büchi-Spiele sind Paritätsspiele mit $c = 2$ und Paritätsspiele mit $c = 3$ sind äquivalent zu 1-Paar Streett-Spielen. Die besten Algorithmen für Mittelwertsspiele laufen in pseudo-polynomialer Zeit $O(mnW)$ [Br11] und in randomisierter sub-exponentieller Zeit $O(2^{\sqrt{n \log n}} \log W)$ [BV07].

Paritätsspiele können in $O(n^c)$ vielen *symbolischen Schritten* gelöst werden wenn eine lineare Anzahl von Mengen gespeichert wird [EL86, Zi98] oder mit $O(n^{c/2+1})$ vielen symbolischen Schritten wenn $O(n^{c/2+1})$ viele Mengen verwendet werden [Br97].

3 Ergebnisse

Wir fassen nun die Ergebnisse der Dissertation zusammen. Die Laufzeiten hier sind teilweise vereinfacht, die tatsächlichen Verbesserungen hängen noch von weiteren Eingabeparametern ab. Alle Algorithmen für ω-reguläre Zielvorgaben berechnen die (quasi-sicheren) Gewinnmengen und können leicht modifiziert werden so dass auch die zugehörigen Gewinnstrategien innerhalb der gleichen Laufzeitschranken berechnet werden. Die Details befinden sich in den entsprechenden Kapiteln der Dissertation [Lo16].

Approximationsalgorithmus für das kleineste mittlere Kreisgewicht. Wir präsentieren den ersten *Approximationsalgorithmus* für *Mittelwerts-Zielvorgaben auf Graphen*, wobei sich die Berechnung der Gewinnmenge auf die Bestimmung des Kreises mit dem kleinstem durchschnittlichem Kantengewicht reduzieren lässt. Dies ist ein grundlegendes graphtheoretisches Problem, das auch Anwendungen bei der Berechnung von Flüssen mit minimalen Kosten in Graphen hat. Im Vergleich zu den lange bekannten exakten Algorithmen, hat der Algorithmus eine Laufzeitschranke mit verbesserter Abhängigkeit von n und ist damit eine Verbesserung für dichte Graphen (das sind in diesem Fall Graphen mit $m = \Theta(n^2)$). Der Algorithmus berechnet für positive Kantengewichte eine multiplikative $(1+\varepsilon)$-Approximation des kleinesten durchschnittlichen Kantengewichts eines Kreises in Zeit $O(n^\omega \log^3(nW/\varepsilon)/\varepsilon)$, wobei $O(n^\omega)$ die beste asymptotische Laufzeit für die Multiplikation zweier $n \times n$ Matrizen ist. Wir reduzieren das Problem zuerst auf die wiederholte Anwendung von min-plus Matrixmultiplikation, die wiederum durch die Verwendung von klassischer Matrixmultiplikation approximiert werden kann. Dieser Ansatz liefert eine bessere asymptotische Laufzeit, die jedoch nicht praxisrelevant ist. Eine interessante zukünftige Forschungsfrage wäre daher die Entwicklung anderer Methoden um die Laufzeit für dieses grundlegende Graphproblem zu verbessern sowie die Erkundung der optimalen Balance zwischen Approximationsgarantie und Laufzeit.

Algorithmen für MEPs mit Streett-Zielvorgaben. Für Streett-Zielvorgaben zeigen wir für MEPs den ersten Algorithmus mit sub-quadratischer Laufzeit sowie einen Algorithmus mit verbesserter Laufzeit für dichte Graphen und MEPs. In ihrer vereinfachten Form sind die Laufzeiten $O(m^{1.5}\sqrt{\log n})$ und $O(n^2)$, was für MEPs die Laufzeit um einen Faktor von $n/\sqrt{\log n}$ bzw. n verbessert und für Graphen die Laufzeit verbessert wenn $m \in \omega(n^{4/3}/\sqrt[3]{\log n})$. Während der einfachste Algorithmus für MEPs mit Streett-

Zielvorgaben bis zu n-mal eine sogenannte Zerlegung in maximale End-Komponenten für ein MEP berechnet, zeigen wir wie die Berechnung von maximalen End-Komponenten durch die einfachere Berechnung von starken Zusammenhangskomponenten ersetzt werden kann. Diese starken Zusammenhangskomponenten können dann wiederum mit graphalgorithmischen Techniken bei Veränderungen des MEPs schneller neu berechnet werden. Dafür verwenden wir eine Sparsifikationstechnik für dichte Graphen um die Laufzeit von $O(n^2)$ zu erhalten sowie einen lokalen Graphexplorationsansatz für die Laufzeit von $O(m^{1.5}\sqrt{\log n})$. Eine offene Fragestellung ist ob die Laufzeit weiter verbessert werden kann oder ob es vielleicht bedingte untere Schranken gibt. Erkenntnisse in jede dieser Richtungen könnten auch zu Fortschritten für andere Graphprobleme führen, für die ähnliche Techniken verwendet wurden.

Paritätsspiele. Für Paritätsspiele zeigen wir zuerst den ersten Algorithmus mit subkubischer Laufzeit von $O(n^{2.5})$ für Paritätsspiele mit drei Prioritäten, was eine Verbesserung der Laufzeit im Fall von $m \in \omega(n^{3/2})$ bedeutet. Zum Zeitpunkt der Dissertation verbesserte dieser Algorithmus die Laufzeit für alle Paritätsspiele mit einer konstanten Anzahl c an Prioritäten, wurde inzwischen aber für $c > 3$ überholt [Fe17]. Während der klassische Algorithmus für Paritätsspiele mit drei Prioritäten wiederholt Büchispiele löst, zeigen wir dass abgeschlossene Teile der Gewinnmenge mit nur $\sqrt{n}$ Knoten bereits in einem Teilgraphen mit nur $O(n^{3/2})$ Kanten gefunden werden können und müssen daher nur für Teile der Gewinnmenge mit mehr als $\sqrt{n}$ Knoten den $O(n^2)$-Zeit Büchispiel-Algorithmus aufrufen.

Weiters zeigen wir einen *symbolischen* Algorithmus, der $O(n^{c/3+1})$ symbolische Schritte benötigt und eine lineare Anzahl von Mengen speichert. Durch eine Variation der Parameter liefert der gleiche Algorithmus auch die erste sub-exponentielle Schranke für die Anzahl der symbolischen Schritte. Dieser Algorithmus verbessert damit die Anzahl der benötigten symbolischen Schritte gegenüber dem bisher bestbekannten symbolischen Algorithmus, während er die gleiche Anzahl an Mengen speichert wie der grundlegende symbolische Algorithmus. Das Kernstück der neuen symbolischen Algorithmen ist eine symbolische Version eines "progress measure" genannten Zählers, der iterativ Daten über das Paritätsspiel sammelt, wobei für die symbolischen Variante $\Theta(n^{c/2})$ viele numerische Werte mit $O(n)$ vielen Mengen repräsentiert werden.

Das große offene Problem für Paritätsspiele ist die Existenz eines Polynomialzeitalgorithmus. Ein weiterer Weg Paritätsspiele besser zu verstehen könnten bedingte untere Schranken sein. Für die praktischen Anwendungen sind symbolische Algorithmen relevant und es wäre interessant ob die neuen Ideen für symbolische Berechnungen auch zu praktischen Verbesserungen führen.

Modell- und Zielvorgaben-Separierung für Graphen und MEPs. Wir zeigen mehrere neue Algorithmen und bedingte untere Schranken für Rabin-Zielvorgaben sowie Disjunktionen von Erreichbarkeits-, Sicherheits-, Büchi- und co-Büchi Zielvorgaben auf Graphen und MEPs. Diese Ergebnisse zeigen zum ersten Mal (1) eine *Separierung der Modelle* und (2) eine *Separierung der Zielvorgaben* für Polynomialzeitprobleme in formaler Verifikation. Für eine Separierung der Modelle, also in diesem Fall von MEPs und Graphen, zeigen

wir für das gleiche algorithmische Problem eine bedingte untere Schranke auf MEPs und eine strikt niedrigere obere Laufzeitschranke auf Graphen und zeigen damit, dass unter den Annahmen (A1) bzw. (A2) das algorithmische Problem auf MEPs schwieriger ist als auf Graphen. Für eine Separierung von Zielvorgaben vergleichen wir auf die gleiche Art und Weise zwei verwandte Zielvorgaben auf dem gleichen Modell. So zeigen wir insbesonders bedingte untere Schranke für Rabin-Zielvorgaben und strikt niedrigere obere Schranken für Streett-Zielvorgaben für sowohl Graphen als auch MEPs.

Die Basis für unsere unteren Schranken sind Reduktionen von CNF-SAT zu Disjunktionen von Erreichbarkeits- und Sicherheits-Zielvorgaben auf MEPs sowie von boolscher Matrixmultiplikation zu Disjunktionen von Sicherheits-Zielvorgaben auf Graphen und zu disjunktiven Erreichbarkeitsabfragen auf MEPs. Wir nützen dann Reduktionen zwischen den verschiedenen Zielvorgaben aus um auch untere Schranken für Büchi-, co-Büchi- und Rabin-Zielvorgaben zu erhalten.

Im Kern der neuen Algorithmen für Disjunktionen von Erreichbarkeits-, Büchi- und co-Büchi-Zielvorgaben auf MEPs stehen Beobachtungen zu den Eigenschaften der maximalen End-Komponenten, welche starke Zusammenhangskomponenten ohne ausgehende Zufallskanten sind. Weiters zeigen wir dass für die Disjunktion von co-Büchi-Zielvorgaben mit nur jeweils einem Knoten in der Zielmenge die Gewinnmenge auf Graphen mit einer Art von Breitensuche in linearer Zeit gelöst werden kann, was zu einer Modellseparierung für dieses Problem führt.

Verallgemeinerte Büchi- und GR(1)-Spiele. *Verallgemeinerte Büchi-Zielvorgaben* sind Konjunktionen von k Büchi-Zielvorgaben und *GR(1)-Zielvorgaben* bestehen aus einer Implikation zwischen zwei verallgemeinerten Büchi-Zielvorgaben. Für verallgemeinerte Büchispiele verbessern wir die Laufzeit für dichte Graphen von $O(k^2 \cdot n^2)$ auf $O(k \cdot n^2)$ und zeigen dass diese Abhängigkeit der Laufzeit von k und n unter der Annahme (A1) optimal ist. Weiters zeigen wir dass der klassische Algorithmus für verallgemeinerte Büchispiele unter der Annahme (A2) eine optimale Abhängigkeit von k und m hat. Diese untere Schranke gilt selbst dann, wenn jede Büchi-Zielmenge nur einen Knoten enthält (in diesem Fall ist die Laufzeit $O(k \cdot m)$). Diese Ergebnisse implizieren weiters eine Modell-Separierung zwischen Spielgraphen einerseits und MEPs und Graphen anderseits. Weiters präsentieren wir einen Algorithmus für GR(1)-Spiele, der die Laufzeit für den Fall $m \in \omega(n^{1.5})$ verbessert.

4 Schlussworte

Diese Dissertation verbindet zwei Teilgebiete der theoretischen Informatik, zwischen denen es viel zu oft kaum Austausch gibt: Algorithmenentwicklung und Komplexitätstheorie auf der einen Seite und Modellprüfung, Automatentheorie und Spielgraphen auf der anderen Seite. In dem wir die asymptotischen Laufzeitschranken von verschiedenen Problemen wie generalisierten Büchispielen mit der von CNF-SAT und kombinatorischer boolescher Matrixmultiplikation verknüpfen, verbinden wir fundamentale algorithmische Probleme der beiden Gebiete der theoretischen Informatik und unsere Ergebnisse zeigen, dass algo-

rithmische Verbesserungen für fundamentale Probleme in formaler Verifikation und Synthese Durchbrüche in der Algorithmenentwicklung bedeuten würden. Weiters zeigen wir die Anwendbarkeit neuester Entwicklungen in der Algorithmenentwicklung und Komplexitätstheorie wie bedingte untere Schranken sowie Techniken aus dem Bereich der Graphalgorithmen für kanonische Probleme der Modellprüfung und Synthese. Aus der Sicht der Algorithmenforschung und Komplexitätstheorie ist unser Beitrag eine zugängliche Exposition wichtiger algorithmischer Probleme in der Sprache von Graphalgorithmen. insbesonders Paritäts- und Mittelwertspiele sind zwei der wenigen "natürlichen" Probleme in NP $\cap$ coNP für die noch kein Polynomialzeitalgorithmus bekannt ist und sie sind daher von größtem Interesse für die Algorithmenforschung und Komplexitätstheorie. Unsere Ergebnisse sind ein erster Schritt um die algorithmische Schwierigkeit von Polynomialzeitproblemen in formaler Verifikation und Synthese zu verstehen, es gibt weiterhin viele interessante offene Probleme, von denen einige in der Dissertation aufgelistet sind [Lo16].

Literaturverzeichnis

[Br97] Browne, A.; Clarke, E. M.; Jha, S.; Long, D. E.; Marrero, W. R.: An Improved Algorithm for the Evaluation of Fixpoint Expressions. Theoretical Computer Science, 178(1-2):237–255, 1997.

[Br11] Brim, L.; Chaloupka, J.; Doyen, L.; Gentilini, R.; Raskin, J.-F.: Faster algorithms for mean-payoff games. FMSD, 38(2):97–118, 2011.

[BV07] Björklund, H.; Vorobyov, S. G.: A combinatorial strongly subexponential strategy improvement algorithm for mean payoff games. Discrete Applied Mathematics, 155(2):210–229, 2007.

[Ca17] Calude, C. S.; Jain, S.; Khoussainov, B.; Li, W.; Stephan, F.: Deciding Parity Games in Quasipolynomial Time. In: STOC. S. 252–263, 2017.

[CH14a] Chatterjee, K.; Henzinger, M.: Efficient and Dynamic Algorithms for Alternating Büchi Games and Maximal End-component Decomposition. Journal of the ACM, 61(3):15, 2014.

[Ch14b] Chatterjee, K.; Henzinger, M.; Krinninger, S.; Loitzenbauer, V.; Raskin, M. A.: Approximating the minimum cycle mean. Theoretical Computer Science, 547:104–116, 2014.

[Ch16a] Chatterjee, K.; Dvořák, W.; Henzinger, M.; Loitzenbauer, V.: Conditionally Optimal Algorithms for Generalized Büchi Games. In: MFCS. S. 25:1–25:15, 2016.

[Ch16b] Chatterjee, K.; Dvořák, W.; Henzinger, M.; Loitzenbauer, V.: Model and Objective Separation with Conditional Lower Bounds: Disjunction is Harder than Conjunction. In: LICS. S. 197–206, 2016.

[Ch17] Chatterjee, K.; Dvořák, W.; Henzinger, M.; Loitzenbauer, V.: Improved Set-Based Symbolic Algorithms for Parity Games. In: CSL. S. 18:1–18:21, 2017.

[CHL17] Chatterjee, K.; Henzinger, M.; Loitzenbauer, V.: Improved Algorithms for Parity and Streett objectives. Logical Methods in Computer Science, 13(3), 2017. Announced at LICS'15.

[CJH03] Chatterjee, K.; Jurdziski, M.; Henzinger, T. A.: Simple stochastic parity games. In: CSL. S. 100–113, 2003.

[EJ88] Emerson, E. A.; Jutla, C. S.: The Complexity of Tree Automata and Logics of Programs (Extended Abstract). In: FOCS. S. 328–337, 1988.

[EL86] Emerson, E. A.; Lei, Ch.-L.: Efficient Model Checking in Fragments of the Propositional Mu-Calculus. In: LICS. S. 267–278, 1986.

[Fe17] Fearnley, J.; Jain, S.; Schewe, S.; Stephan, F.; Wojtczak, D.: An Ordered Approach to Solving Parity Games in Quasi Polynomial Time and Quasi Linear Space. In: SPIN. S. 112–121, 2017.

[FV97] Filar, J.; Vrieze, K.: Competitive Markov Decision Processes. Springer-Verlag, 1997.

[HT96] Henzinger, M.; Telle, J. A.: Faster Algorithms for the Nonemptiness of Streett Automata and for Communication Protocol Pruning. In: SWAT. S. 16–27, 1996.

[JPZ08] Jurdziński, M.; Paterson, M.; Zwick, U.: A Deterministic Subexponential Algorithm for Solving Parity Games. SIAM J. Comput., 38(4):1519–1532, 2008.

[Ju00] Jurdziński, M.: Small Progress Measures for Solving Parity Games. In: STACS. S. 290–301, 2000.

[Ka78] Karp, R. M.: A characterization of the minimum cycle mean in a digraph. Discrete Mathematics, 23(3):309–311, 1978.

[Lo16] Loitzenbauer, V.: Improved Algorithms and Conditional Lower Bounds for Problems in Formal Verification and Reactive Synthesis. Dissertation, University of Vienna, 2016.

[OA92] Orlin, J. B.; Ahuja, R. K.: New scaling algorithms for the assignment and minimum mean cycle problems. Mathematical Programming, 54(1-3):41–56, 1992.

[Sc07] Schewe, S.: Solving Parity Games in Big Steps. In: FSTTCS. S. 449–460, 2007.

[Zi98] Zielonka, W.: Infinite games on finitely coloured graphs with applications to automata on infinite trees. Theoretical Computer Science, 200(1–2):135–183, 1998.

Veronika Loitzenbauer, geboren 1988, hat nach dem Bachelorstudium Computational Science an der Karl-Franzens Universität Graz und dem Masterstudium Scientific Computing an der Universität Wien ihre Leidenschaft für theoretische Informatik und insbesondere Graphalgorithmen entdeckt. Sie hat von 2012 bis 2017 an der Universität Wien in der Forschungsgruppe „Theorie und Anwendung von Algorithmen“, betreut von Prof. Monika Henzinger, promoviert. In dieser Zeit hat sie sowohl an algorithmischer Spieltheorie, fundamentalen Graphproblemen, als auch an denen in dieser Dissertation präsentierten algorithmischen Problemen in formaler Verifikation und Synthese geforscht. Ihre Dissertation wurde mit dem österreichischen Staatspreis „Award of Excellence“ ausgezeichnet. Sie hat Forschungsaufenthalte an der Università di Roma „Tor Vergata“, Italien, der University of Michigan, USA, und der Bar-Ilan Universität, Israel, absolviert. Seit Dezember 2017 forscht sie als PostDoc an der Johannes Kepler Universität Linz am Institut für formale Modelle und Verifikation.