

Stern-Topologie-Entkoppelte Zustandsraumsuche in der KI-Planung und Modellprüfung¹

Daniel Gnad²

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 eingeführt, 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 $\mathcal{L} = \{L_1, \dots, L_n\}$, 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 $\Pi = \langle \mathcal{V}, \mathcal{A}, \text{cost}, \mathcal{I}, \mathcal{G} \rangle$, 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

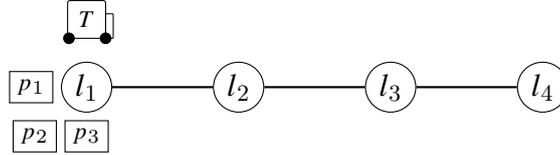


Abb. 1: Eine grafische Illustration des initialen Zustands unseres Beispiels.

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

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

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

Domäne	#	# \mathcal{F}	Blinde Suche							A^* mit h^{LM-cut}							sbd	c2
			Explizite Suche							Explizite Suche								
			unf	b	pp	por	sym	p+s	DS	b	pp	por	sym	p+s	DS			
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^{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-

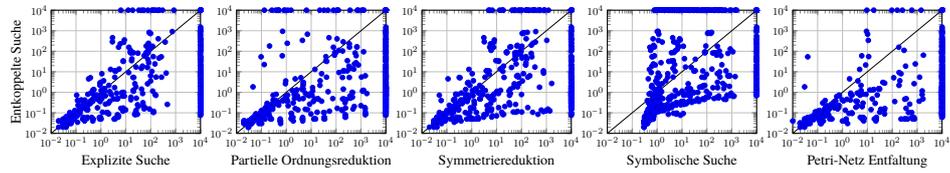


Abb. 3: Vergleich der Laufzeit (in Sekunden) zum kompletten Aufbau des Zustandsraums von entkoppelter Suche (DS) auf der y-Achse und verschiedenen alternativen Methoden auf der x-Achse.

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^{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.

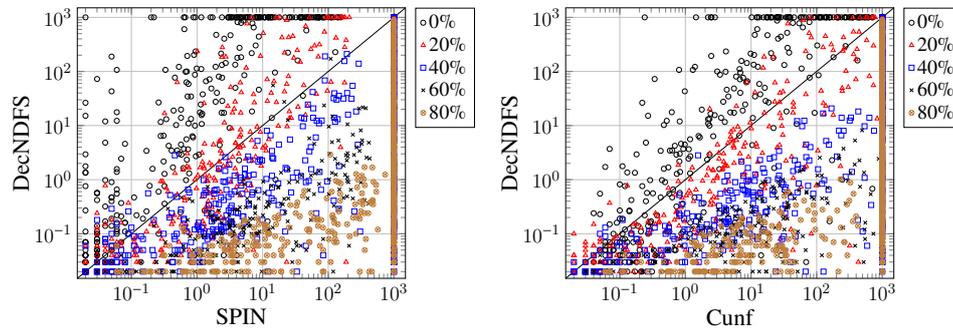


Abb. 4: Laufzeitstatistik, je Probleminstanz, für die Prüfung von Lebendigkeitseigenschaften von entkoppelter Suche, DecNDFS, auf der y -Achse, gegen SPIN (links) und Cunft (rechts), jeweils auf der x -Achse. Farblich hervorgehoben ist der Anteil an internen Transitionslabels.

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 Erreichbarkeitsprüfung 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 Cunft Petri-Netz Entfaltungs-tool [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* [A112], 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] Alkharaji, 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 maschinellen 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.