

Kurzbericht

# NAVAS - Navigation Approaches for Answer Sets

Navigation im Lösungsraum von Answer Sets

Sarah Alice Gaggl

## 1 Ursprüngliche Aufgabenstellung

Durch die kontinuierliche Weiterentwicklung in den letzten Jahren können heute eine Vielzahl anspruchsvoller kombinatorischer Suchprobleme von enormer praktischer Relevanz effizient durch Answer Set Programming (ASP) gelöst werden. Dabei kann es je nach Umfang des zu lösenden Problems sehr rasch zu einer “kombinatorischen Explosion” der Menge von Lösungen kommen. Während moderne ASP Solver problemlos mehrere Millionen von Lösungen in kurzer Zeit berechnen können, stellt dies den Anwender vor ein neues Problem: Wie soll der enorme Lösungsraum gehandhabt und zugänglich gemacht werden?

Typischer Weise werden die Answer Sets ungeordnet ausgegeben. In der Praxis sind jedoch oft nur Lösungen von Interesse, die sich hinreichend voneinander unterscheiden, die besondere Eigenschaften aufweisen, oder die einer gegebenen Menge von Lösungen ähnlich sind. Das Projekt NAVAS befasste sich mit den folgenden Themen:

1. Entwicklung von Methoden zur interaktiven und transparenten Navigation im Lösungsraum der Answer Sets;
2. Implementierung der Methoden mit effizienten Algorithmen;
3. Evaluation des Prototyps anhand von Anwendungsfällen aus den Bereichen Konfiguration und Argumentation.

Wir sind zuversichtlich, dass ASP-Technologie durch die in NAVAS erzielten Erkenntnisse, Entwicklungen und Implementierungen auch für eine breitere Masse zugänglicher wird. Hierbei sehen wir ein besonderes unmittelbares Potential im Bereich industrieller Anwendungen, etwa zur Produktkonfiguration.

## 2 Ablauf

Das Projekt war in 6 Arbeitspakete unterteilt, die teilweise aufeinander aufbauten.

AP1 Theoretische Grundlagen

AP2 Metriken und Klassifikatoren

AP3 Einschränkungen des Suchraums

AP4 Implementierung

AP5 Anwendungsfälle

AP6 Evaluation

Diese wurden von PI Gaggl und den Mitarbeiter:Innen Elisa Böhl und Dominik Rusovac bearbeitet.

## 3 Wesentliche Ergebnisse des Vorhabens

Im Zuge des Projekts wurden alle Arbeitspakete bearbeitet und erfolgreich Methoden entwickelt um in dem ASP Lösungsraum zu navigieren. Besonderes Augenmerk wurde auf die Entwicklung von Methoden gelenkt, die selbst in Lösungsräumen skalieren, die zu groß sind um sie enumerieren zu können. Dies ist uns mit mehreren Ansätzen gelungen. Trotzdem bleibt das zu lösende Problem äußerst komplex und schwierig.

Die Thematik der großen Lösungsräume kommt in vielen Bereichen vor, daher ist anzunehmen, dass das Thema weiterhin große Bedeutung findet, und die erzielten Resultate eine gute Grundlage für weitere Entwicklungen bildet.

Es entstanden 24 Publikationen, darunter 4 Journalpublikationen, 14 Konferenzpublikationen, 3 Technische Berichte, 3 herausgegebene Tagungsbände, 6 eingeladene Vorträge, und eine Systembeschreibung durch die Teilnahme an dem ICCMA 2023 Wettbewerb (siehe [https://iccl.inf.tu-dresden.de/web/NAVAS/Publication\\_report/en](https://iccl.inf.tu-dresden.de/web/NAVAS/Publication_report/en)). Es gab eine abgeschlossene Dissertation (Verteidigung 14.5.2025), 3 abgeschlossene Master Arbeiten und eine abgeschlossene Bachelor Arbeit (siehe [https://iccl.inf.tu-dresden.de/web/Logische\\_Programmierung\\_und\\_Argumentation/Abgeschlossene\\_Studienarbeiten](https://iccl.inf.tu-dresden.de/web/Logische_Programmierung_und_Argumentation/Abgeschlossene_Studienarbeiten)).

Prototypen sind hier erhältlich:

- fasb <https://github.com/drwadu/fasb>
- web-tool von fasb <https://drwadu.github.io/web-fasb.github.io/>
- iascar <https://github.com/drwadu/iascar>
- tunas <https://doi.org/10.5281/zenodo.6762554>
- snake <https://github.com/elbo4u/asp-snake-ms>
- web-tool nexas: <https://imld.de/en/research/research-projects/nexas/>

Sarah Alice Gaggl  
Nöthnitzer Straße 46  
01187 Dresden  
+49(0)351 463 38679  
sarah.gaggl@tu-dresden.de

## **Schlussbericht**

# **NAVAS - Navigation Approaches for Answer Sets**

**Navigation im Lösungsraum von Answer Sets**

Sarah Alice Gaggl

Das diesem Bericht zugrunde liegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IS20056 gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt bei der Autorin.



# 1 Ursprüngliche Aufgabenstellung

Ein zentrales traditionelles Forschungsfeld der Künstlichen Intelligenz ist die effiziente Suche nach Lösungen komplexer Probleme, welche eine Reihe formal spezifizierter Anforderungen (so genannter “Constraints”) erfüllen. Eine Vielzahl auf den ersten Blick sehr verschiedenartiger anspruchsvoller Aufgaben lassen sich einheitlich als solche “constraint satisfaction problems” darstellen: Konfiguration komplexer Systeme, Produkte und Softwaresysteme, Planung miteinander verwobener Prozesse, Optimierung logistischer Abläufe, Handhabung von Abhängigkeiten in Softwaresystemen sind nur einige Beispiele, anhand derer die eminente Bedeutung intelligenter Unterstützung bei der Beherrschung von Situationen klar wird, deren Komplexität über das vom Menschen Erfassbare hinausgeht. Zu welcher wirtschaftlich und gesellschaftlich dramatischen Folgen die Abwesenheit solcher intelligenter Technologien führen kann, ist in den letzten Jahren auch in Deutschland mehrfach deutlich geworden, hier seien nur Stuttgart 21, die Elbphilharmonie, oder der Flughafen Berlin Brandenburg als besonders drastische Beispiele genannt.

Einen sehr vielversprechenden, in Pilotprojekten bereits praktisch vielfach bewährten Ansatz zur generischen Modellierung und Lösung von derartigen Problemen stellt das *Answer Set Programming* dar. Durch die kontinuierliche Verbesserung und Weiterentwicklung von Werkzeugen zur Lösung von Answer-Set-Programmen (“ASP Solvern”) in den letzten Jahren [Gebser et al., 2017] kann mittlerweile eine Vielzahl anspruchsvoller kombinatorischer Suchprobleme effizient gelöst werden. Dabei kann es – zum Beispiel bei Konfigurationsproblemen – je nach Umfang des zu lösenden Problems sehr rasch zu einer “kombinatorischen Explosion” der Menge von Lösungen kommen. Für moderne ASP Solver ist es ein Leichtes, mehrere Millionen von Lösungen (so genannte *Answer Sets*) in kurzer Zeit zu berechnen. Das folgende ASP Programm `grid.lp` stellt beispielsweise ein stark vereinfachtes Konfigurationsproblem dar: auf einem Raster der Größe  $n \times m$  sollen  $k \geq n \times m$  Objekte so angeordnet werden, dass jedes Objekt nur einmal eingesetzt wird.

```
cell(1..U) :- U=n*m.  
obj(1..k).  
1{set_obj_cell(X,C):obj(X)}1 :- cell(C).  
:- set_obj_cell(X,C1), set_obj_cell(X,C2), C1=C2.
```

Aufgerufen mit  $n = 3$ ,  $m = 3$  und  $k = 9$

```
$ clingo grid.lp --const n=3 --const m=3 --const k=9 0
```

berechnet der populäre ASP Solver `clingo` [Gebser et al., 2011] 362.889 Answer Sets in nur 0,4 Sekunden. Wenn man  $k$  auf 11 erhöht bei gleichbleibender Größe des Rasters, sind es bereits 3.628.800 Answer Sets, erzeugt in nur 3,25 Sekunden. Mit  $k = 15$  erhalten wir bereits 1.816.214.400 Answer Sets in 1703,97 Sekunden.

## 1.1 Gesamtziel des Vorhabens

Auch wenn das oben genannte Beispiel sehr vereinfacht ist und durch die geringen Einschränkungen an die Konfiguration selbst für kleine Raster sehr viele Lösungen hat, ist es klar ersichtlich, dass man in praktischen Anwendungen ebenfalls sehr rasch einen Lösungsraum von mehreren Millionen Answer Sets erhalten kann. Das stellt jedoch den Anwender

vor ein neues Problem: Wie soll der enorme Lösungsraum gehandhabt werden?

Typischer Weise werden die Answer Sets in beliebiger Reihenfolge ausgegeben. Jedoch sind viele dieser Lösungen sehr ähnlich. Zur Veranschaulichung zeigen die Raster a) und b)

4	3	1
7	8	5
6	9	2

4	3	1
8	7	5
6	9	2

13	2	5
1	11	10
3	7	9

10	4	3
6	9	8
1	5	7

Abbildung 1: Raster a) bis d) von links nach rechts.

in Abbildung 1 zwei sehr ähnliche Lösungen zu dem Beispiel, wohingegen Raster c) und d) keine Gemeinsamkeiten mit allen anderen haben.

In der Praxis sind jedoch oft nur Lösungen von Interesse, die sich hinreichend voneinander unterscheiden, die besondere Eigenschaften aufweisen, oder die einer gegebenen Menge von Lösungen ähnlich sind. Derlei Features werden allerdings bis jetzt noch nicht von ASP Solvern unterstützt.

Ziel von NAVAS ist es, der Nutzerin und dem Nutzer eine intuitive Navigation im Lösungsraum von ASP-Programmen zu ermöglichen. Der ASP-Solver soll nicht länger eine *black box* sein, die unkontrolliert und ungeordnet Lösungen ohne eine Möglichkeit der Einflussnahme produziert. Wir möchten vielmehr eine interaktive, transparente und zielgerichtete Navigation zu genau den Gebieten im Lösungsraum ermöglichen, die für die Anwendung von besonderer Bedeutung und Interesse sind. Dabei soll das ASP-Programm an sich nicht *korrigiert* werden, denn im Gegensatz zu existierenden *ASP-Debugging-Ansätzen* gehen wir hier nicht von der Annahme aus, dass manche Lösungen falsch sind. Vielmehr ist der Lösungsraum an sich so groß, dass nicht alle Answer Sets weiterverwendet oder bearbeitet werden können oder sollen. Der Anwenderin und dem Anwender sollen jedoch sowohl ein *Überblick* als auch ein genauer *Einblick* in diesen Lösungsraum ermöglicht werden, indem dieser unter unterschiedlichen Gesichtspunkten analysiert wird. Dabei werden die weiter unten aufgeführten Navigations-Strategien zum Einsatz kommen.

## 1.2 Bezug zum Förderprogramm

Das Vorhaben ist klar dem Bereich Grundlagen der KI zuzuordnen, insbesondere zu den Teilbereichen Wissensrepräsentation und Explainable AI. Answer Set Programming ist ein etablierter Formalismus im Bereich Wissensrepräsentation und -verarbeitung. Das Hauptziel des Vorhabens ist es, den ASP Lösungsraum transparenter und dadurch auch erklärbarer zu machen. Mit Hilfe der zu entwickelnden Navigationsstrategien werden neue Wege in der Repräsentation von ASP Lösungen ermöglicht.

## 1.3 Wissenschaftliche und technische Arbeitsziele

Um eine interaktive und transparente Navigation in ASP-Lösungsräumen zu ermöglichen, wird sich das Projekt NAVAS mit den folgenden Themen befassen:

1. Entwicklung und Untersuchung unterschiedlicher Methoden zur Navigation im Lösungsraum der Answer Sets;

2. Implementierung der Methoden mit effizienten Algorithmen;
3. Evaluation des daraus resultierenden Prototyps bezüglich seiner Leistungsfähigkeit anhand von Anwendungsfällen aus den Bereichen Konfiguration und Argumentation.

Wir sind zuversichtlich, dass ASP-Technologie durch die in NAVAS erzielten Erkenntnisse, Entwicklungen und Implementierungen auch für eine breitere Masse von Anwenderinnen und Anwendern zugänglicher wird. Hierbei sehen wir ein besonderes Potential im Bereich industrieller Anwendungen, etwa zur Produktkonfiguration.

## 2 Stand der Wissenschaft und Technik

Answer Set Programming (ASP) ist ein aus der Logikprogrammierung hervorgegangener mächtiger Formalismus zur Wissensrepräsentation und -verarbeitung [Bonatti et al., 2010, Anger et al., 2005, Brewka et al., 2011]. Den Anfang von ASP markiert die Einführung der Stable Model Semantics in den 80er Jahren durch Gelfond und Lifschitz [Gelfond and Lifschitz, 1988, 1991]. Seitdem erfreut sich das Forschungsfeld einer kontinuierlich steigenden Popularität mit einer Vielzahl von Publikationen bei hochrangigen Konferenzen (ICLP, LP-NMR, IJCAI, ECAI und viele mehr), sowie in renommierten Journalen wie AIJ, JAIR, TPLP, etc. Der Erfolg von ASP lässt sich auf die folgenden Punkte zurückführen: (i) Der deklarative Ansatz erlaubt die Modellierung einer Vielzahl von Problemen auf intuitive und modulare Art und Weise [Gebser et al., 2012, Cerutti et al., 2018, Charwat et al., 2015, Egly et al., 2010, Gaggl and Woltran, 2013]; (ii) Die ausdrucksstarke Sprache ermöglicht es sogar Probleme bis zur hohen Komplexität von  $\Sigma_2^P$  auszudrücken [Gaggl et al., 2015a, Dvořák et al., 2013]; (iii) Die kontinuierliche Weiterentwicklung von leistungsstarken ASP-Solvern [Gebser et al., 2017] hat dazu geführt, dass ASP mittlerweile auch in praktischen Bereichen zur Anwendung kommt, wie zum Beispiel Konfiguration, Planung, Robotik, Scheduling, Diagnose [Erdem et al., 2016, Erdem and Patoglu, 2018, Kahraman and Erdem, 2019, Gençay et al., 2019, Falkner et al., 2018].

Durch die Vereinigung von Techniken aus Wissensrepräsentation und -verarbeitung sowie SAT Solving bietet ASP eine hervorragende Basis um kombinatorische Suchprobleme zu modellieren. Dabei besteht ein ASP-Programm aus einer Menge von logischen Regeln, die auch *negation as failure* (Negation durch Fehlschlag) enthalten können. Basierend auf der *closed world assumption* werden dann alle Stable Models, auch Answer Sets genannt, berechnet. Durch den nicht-monotonen Lösungsansatz können insbesondere viele Probleme von hoher praktischer Relevanz modelliert werden, wie weiter oben schon erwähnt. Aber auch für die Entwicklung von Prototypen zur Lösung von eher wissenschaftlich motivierten Problemen wie in Abstract Argumentation [Dung, 1995], kommt dieser *reduction-based* Ansatz häufig zum Einsatz [Cerutti et al., 2017, 2018]. Dabei bedeutet reduction-based, dass das gegebene Problem auf ein anderes Problem reduziert wird, um bereits existierende Solver für letzteres verwenden zu können. Ein Beispiel für einen solchen ASP-basierten Solver für Probleme aus dem Bereich Argumentation ist der Solver ASPARTIX [Gaggl et al., 2015a, Dvořák et al., 2015, 2013, Egly et al., 2010, 2008], der bei vergangenen und aktuellen Wettbewerben (ICCMA 2015 und 2019) [Bistarelli et al., 2018, Thimm et al., 2016, Thimm and Villata, 2017] unter den Top-Plätzen gelistet ist bzw. als Referenzsystem (ICC-

MA 2017) [Gaggl et al., 2018, 2016a] fungiert hat. Ein weiteres prominentes Beispiel für die Reduktion auf ASP ist das sogenannte *Fixed-Domain Reasoning* für Beschreibungslogiken und Ontologiesprachen, wobei alle Modelle einer Wissensbasis bzw. Ontologie über einem fixierten Gegenstandsbereich zu berechnen sind [Gaggl et al., 2016b].

Zur Berechnung der Lösung(en) eines ASP-Programms wird dieses einem ASP Solver übergeben, der je nach Aufruf ein, mehrere oder alle Answer Sets berechnet. Dabei wird in einem ersten Schritt das sogenannte *Grounding* erstellt (zumeist intern), wobei im wesentlichen alle im ASP-Programm vorkommenden Variablen auf alle möglichen Weisen instanziiert werden. Intelligente Groundingstrategien [Kaufmann et al., 2016] sorgen dafür, dass nur solche Groundinstanzen von Regeln erstellt werden, die später relevant werden können. Nichtsdestotrotz erzeugt das Grounding bei vielen praktischen Problemen bereits eine sehr umfangreiche Ausgabe und beansprucht auch schon einen großen Teil an Rechenzeit und Speicher beim Lösungsvorgang. Hier setzt der relativ neue Ansatz des *lazy groundings* an, bei welchem nicht das gesamte Programm gegroundet wird, sondern nur gewisse Teile [Taupe et al., 2019, Eiter et al., 2018]. Das gegroundete Programm wird dem Solver zum Lösen übergeben. Dabei kommen besonders Techniken die sich beim *SAT Solving* bewährt haben zum Einsatz: mit Hilfe des *Conflict Driven Nogood Learning (CDNL)* Algorithmus (siehe [Gebser et al., 2012] für eine detaillierte Beschreibung) werden dann die Answer Sets berechnet.

Weiterentwicklungen der letzten Jahre wie Integration von Skriptsprachen wie Lua und Python, *dlvhex*, Multi-Shot ASP-Solving, Erweiterungen um Präferenzen und temporale Operatoren [Brewka et al., 2015, Adrian et al., 2018, Gebser et al., 2019, Cabalar et al., 2019] machen ASP zu einem immer vielseitigeren und mächtigeren Entwicklungsansatz.

Moderne ASP-Solver wie der Open Source Solver *clingo* [Gebser et al., 2011] oder das kommerzielle System *dlv* [Leone et al., 2006] können, je nach Art des Problems, mehrere Millionen Answer Sets in wenigen Sekunden berechnen. Bei der darauffolgende Erschließung des erzeugten gewaltigen Lösungsraums bleibt der Anwender jedoch vollkommen auf sich gestellt. Dazu müssten alle Answer Sets im Nachhinein analysiert und gegebenenfalls miteinander verglichen werden, was schon durch deren große Anzahl nichttrivial ist. Die typischen Inferenzprobleme für ASP sind das Zählen von Answer Sets sowie *brave* und *cautious reasoning*. Brave reasoning beantwortet die Frage, ob ein Atom (also eine elementare Aussage) in allen Answer Sets enthalten ist (d.h., notwendigerweise in allen Lösungen gilt), wohingegen ein Atom *cautiously entailed* ist, wenn es in mindestens einem Answer Set vorkommt.

Erste Ansätze, um den großen Lösungsraum einzuschränken, wurden kürzlich in [Al-Rabaa et al., 2018] präsentiert. Dieser stellt eine *Facettenklassifikation* dar, wodurch man schrittweise zu dem gewünschten Answer Set navigieren kann. Weiters wird eine uneingeschränkte Anwendung von Facetten vorgestellt, die auch zu Widersprüchen zwischen den Facetten führen kann. Diese Arbeit ist für uns von großer Relevanz, und soll auch weitergeführt werden, siehe *Faceted Browsing* in **AP 3(a)**.

Die einzigen uns bekannten Arbeiten über Vergleiche von Answer Sets bezüglich ihrer Ähnlichkeit beziehungsweise Unterschiedlichkeit wurden in [Eiter et al., 2009, 2013, Romero et al., 2016] präsentiert. In [Eiter et al., 2009, 2013] werden 4 Methoden zur Berechnung unterschiedlicher Answer Sets verglichen. Dabei wird jedoch die Definition eines Maßes für

die Unterschiedlichkeit der Anwenderin oder dem Anwender überlassen, bzw. ist stark vom Anwendungsfall abhängig. Außerdem wurden die Methoden nicht auf ihre Performanz hinsichtlich großer Mengen von Answer Sets untersucht. In [Romero et al., 2016] werden diese Techniken für Programme mit Präferenzen generalisiert.

Ein auf den ersten Blick ähnlich scheinendes Forschungsgebiet ist das Debugging von ASP-Programmen [Oetsch et al., 2018, Busoniu et al., 2013, Vos et al., 2012, Dodaro et al., 2019, Gasteiger et al., 2016, Shchekotykhin, 2015]. Hier ist jedoch klar zu unterscheiden, dass man beim Debugging davon ausgeht, dass das ASP Programm fehlerhaft ist, somit die Menge an berechneten Answer Sets nicht korrekt ist.

Im Bereich Visualisierung von Answer Sets sind die folgenden Systeme erwähnenswert: Kara [Kloimüller et al., 2011], ASPIDE [Febbraro et al., 2011] und ARVis [Ambroz et al., 2013]. Jedoch ist keines dieser Systeme auf die Visualisierung von großen Mengen von Answer Sets ausgerichtet.

**Eigene Vorarbeiten.** Die Antragsstellerin verfügt über umfassende Erfahrung in der Entwicklung von ASP-basierten Softwaresystemen. Besonders im Bereich Abstract Argumentation hat das von Frau Gaggl mitentwickelte System ASPARTIX [Egly et al., 2010, Gaggl et al., 2015a] einen großen Einfluss auf die automatische Evaluierung von Argumentation Frameworks und in weiterer Folge auf die Entwicklung von Solvern, für deren Evaluierung die International Competition for Computational Models of Argumentation (ICCMA) im Jahr 2019 bereits in der dritten Ausgabe gesorgt hat. Eine ausführliche Beschreibung von Methoden zur Lösung von Inferenzproblemen von Abstract Argumentation Frameworks wird in einem Übersichtsartikel gegeben [Charwat et al., 2015]. Weiters hat die Antragstellerin im *Handbook on Formal Argumentation* das Kapitel *Foundations of implementations for formal argumentation* mitverfasst [Cerutti et al., 2018].

Für Abstract Dialectical Frameworks (ADFs), einer Generalisierung von Abstract Argumentation Frameworks, konnte mit einem rekursiven Zerlegungsschema der Grundstein für weitere effizientere Implementierungsmethoden von ADF-Semantiken gelegt werden [Gaggl and Strass, 2014]. Die daraus neu resultierenden Semantiken *cft* und *stg2* für ADFs wurden zusammen mit weiteren naive-based Semantiken ausführlich bezüglich ihrer Komplexität untersucht [Gaggl et al., 2015c].

Im Zuge von Arbeiten zu Fixed-domain reasoning, die aus einem Industrieprojekt heraus inspiriert wurden [Gaggl et al., 2016b, 2015b], stellte sich schnell heraus, dass bei Konfigurationsproblemen der Lösungsraum Größen annimmt, die mit bekannten Methoden nicht mehr zu handhaben sind. Diese Erkenntnis und die Erfahrungen die während der Organisation und Durchführung der ICCMA 2017 [Gaggl et al., 2018, 2020] gesammelt wurden, haben gezeigt, dass in diesem Bereich noch viel Handlungsbedarf und Potential liegt, und führten in weiterer Folge zu der Idee des Projektes NAVAS.

### 3 Ablauf des Vorhabens

Um eine Navigation im Answer Set Raum zu ermöglichen, wurden die nachfolgend beschriebenen Strategien entwickelt und in **AP 4** prototypisch implementiert. Die Analyse war nicht

zwangsläufig auf die im Projektplan aufgeführten Strategien beschränkt. Diese wurden im Zuge des Projektes erweitern, modifizieren und auch miteinander kombinieren.

### 3.1 Vorhabenbezogene Arbeits- und Ressourcenplanung

Das Vorhaben war in 6 Arbeitspakete unterteilt, die aufeinander aufbauen und im Folgenden vorgestellt werden. In jedem Arbeitspaket werden die Ergebnisse gelistet.

#### Arbeitspaket 1 – Theoretische Grundlagen

Als Basis und Input für nachfolgende Arbeitspakete müssen zuerst die Notationen fixiert und passende Methoden ausgewählt werden. Eine umfassende Komplexitätsanalyse der in **AP 2** und **AP 3** definierten Methoden ist unumgänglich, um diese in **AP 4** adäquat zu implementieren.

**AP 1(a): Notationen und Methoden.** Als Grundlage für alle weiteren Schritte müssen die grundlegenden Termini, neue Begrifflichkeiten (etwa zur strukturellen Charakterisierung des Lösungsraumes), sowie die zu untersuchenden Strategien und deren Eigenschaften formal definiert werden. Dazu benötigt man ein einheitliches notationelles Rahmenwerk, welches die weiteren konzeptionellen Schritte vereinfacht, wie zum Beispiel die Kombination von zwei oder mehreren Strategien. Grundlage für die ASP-eigenen Notationen wird der ASP Core 2 sein [Calimeri et al., 2012].

**Ergebnisse:** Dieses Arbeitspaket wird in fast allen Arbeiten behandelt, wie aus der Tabelle der Abhängigkeiten der Arbeitspakete ersichtlich ist. Daher werden die Ergebnisse nicht einzeln angeführt.

**AP 1(b): Komplexitätsanalyse.** Eine umfassende Komplexitätsanalyse der angewandten Methoden ist essenziell für die Entwicklung und Auswahl von effizienten Algorithmen und Implementierungsmethoden in **AP 4**. Da das Ziel dieses Vorhabens die Navigation in besonders großen Lösungsräumen war, müssen die einzelnen Navigationsstrategien von **AP 2** und **AP 3** nach komplexitätstheoretischen Aspekten analysiert werden, um deren prinzipielle Skalierbarkeit verlässlich abschätzen zu können. Die Ergebnisse dieser Analyse lieferten wichtige Einsichten für die Wahl der Implementierungsmethode in **AP 4**.

**Ergebnisse:** In [Rusovac et al., 2024] wurde die Komplexität der entsprechenden reasoning Probleme zu den zugehörigen counting Problemen analysiert. Die Arbeit wurde auf der KR 2024 publiziert und präsentiert. Weitere Komplexitätsresultate zu den entwickelten Methoden sind in den jeweiligen Publikationen enthalten [Fichte et al., 2022b, Böhl and Gaggl, 2022]

#### Arbeitspaket 2 – Metriken und Klassifikatoren

In diesem Arbeitspaket werden Metriken und Klassifikatoren für Answer Sets definiert und untersucht. Durch die Auswahl von *Metriken* (Ähnlichkeitsmaßen) sollen Answer Sets hinsichtlich verschiedener Kriterien miteinander verglichen werden können. Damit soll es ermöglicht werden, möglichst unterschiedliche bzw. ähnliche Answer Sets zu berechnen. Die in **AP 2(a)** und **2(b)** entwickelten Navigationsstrategien werden von diesen Metriken Gebrauch machen.

In **AP 2(c)** hingegen soll ein *Klassifikator* gelernt werden, um zu dem gewünschten Answer Set zu gelangen.

**AP 2(a): Punktweise Navigation.** Mit Hilfe der Definition von Metriken und Nachbarschaftsbegriffen für Answer Sets werden diverse Kriterien für Unterschiedlichkeit bzw. Ähnlichkeit von Lösungen untereinander formal charakterisiert. Mithilfe dieser topologischen Begrifflichkeiten soll nun “punktweise” im Lösungsraum navigiert werden, von einem Answer Set zum “nächsten”, abhängig von der vorab definierten Metrik oder Nachbarschaft. Diese Navigation kann insbesondere interaktiv während der Anwendung angepasst werden, indem gewisse Kriterien, die von den Answer Sets erfüllt werden müssen, während der Navigation bestimmt werden können.

**Ergebnisse:** Dieses Arbeitspaket wurde mit **AP 2(b)** zusammengelegt und die Metriken wurden in den dort entwickelten Ansätzen definiert. Die Punktweise Navigation ist ein Spezialfall des Answer Set Samplings, daher verweisen wir auf die Ergebnisse aus **AP 2(b)**.

**AP 2(b): Answer Set Sampling.** Hier soll mit Hilfe der vorab definierten Metriken (siehe Punktweise Navigation) eine Stichprobe von  $N$  möglichst unterschiedlichen Answer Sets berechnet werden. Dadurch soll ein Überblick über die Vielfältigkeit des Lösungsraums gegeben werden. Diese Methode wird sowohl offline als auch online umgesetzt werden, wobei für letzteres ein Eingriff in den Lösungsprozess erforderlich ist.

**Ergebnisse:** Es wurde ein Ansatz entwickelt, der iterativ Kollektionen von Answer Sets berechnet, die das definierte Diversitätsmaß erfüllen. Dabei wird versucht auch wieder Answer Sets zu entfernen um im nächsten Schritt die Kollektion zu verbessern. Die Arbeit wurde auf der LPNMR 2022 publiziert und präsentiert [Böhl and Gaggl, 2022]. Im Anschluss daran wurde der Begriff Representative Answer Sets definiert auf Basis der Entropie. Dazu wurden unterschiedliche Ansätze entwickelt um repräsentative Kollektionen von Answer Sets zu berechnen. Hier wurden auch Ergebnisse von **AP 3(a)** Fichte et al. [2022b] verwendet um die Suche nach repräsentativen Answer Sets zu verbessern. Die Arbeit wurde auf der ECAI 2023 publiziert und präsentiert [Böhl et al., 2023]. Einer weitere Methode zum Sampling wurde in [Rusovac et al., 2024] vorgestellt. Dabei ist es möglich eine Query/Anfrage an das Programm zu stellen, und dann alle Lösungen zu erhalten die der Query entsprechen.

**AP 2(c): Empfehlung von Answer Sets.** Durch Erlernen von Klassifikatoren, basierend auf vorher durch den Nutzer ausgewählten bzw. präferierten Answer Sets, können dann im Sinne eines Recommender Systems weitere vorgeschlagen werden. Dadurch kann sich die Auswahl am Verhalten des Anwenders orientieren, ohne dass selbiger seine Präferenzen explizit spezifizieren muss.

**Ergebnisse:** dieses Arbeitspaket wurde nicht behandelt. Durch Einschränkungen in der Pandemie mussten Prioritäten gesetzt werden und andere Arbeitspakete wie **AP 2(b)** und **AP 3(a)** wurden vertieft.

### Arbeitspaket 3 – Einschränkung des Suchraums

Durch Einschränkungen im Lösungsraum wird die Suche nach den gewünschten Answer Sets zielgerichtet und effizient durch die in **AP 3(a)** und **AP 3(b)** beschriebenen Methoden ermöglicht.

**AP 3(a): Faceted Browsing.** Durch Fixierung einzelner Atome hinsichtlich ihres (Nicht-)Enthaltens im gesuchten Answer Set wird der Lösungsraum eingeschränkt. Nach einem solchen Schritt befindet sich der Anwender nur noch in einem Teilraum, der dann weiter eingeschränkt werden kann. Dadurch wird eine schrittweise Annäherung der Lösung durch iteratives, interaktives Verkleinern des Suchraumes realisiert. Ein erster Ansatz in diese Richtung wurde in [Al-Rabaa et al., 2018] vorgestellt.

**Ergebnisse:** Der Ansatz in [Al-Rabaa et al., 2018] wurde substantiell ausgebaut, indem Gewichte zu den Facetten hinzugefügt werden, um die Suche zu quantifizieren. Dies ermöglicht es zu erkennen welchen Einfluss die Aktivierung einer Facette hat, zum Beispiel wie sehr sie den weiteren Lösungsraum bezüglich der Größe oder Diversität der Lösungen einschränkt. Die daraus resultierende Publikation [Fichte et al., 2022b] wurde auf der AAAI 2022 publiziert. Für die genannten Gewichte ist es notwendig Lösungen, insbesondere Answer Sets zu zählen. Da dies besonders schwer ist (siehe Komplexitätsanalyse) wurde eine Methode des inkrementellen Zählens entwickelt [Fichte et al., 2022a] und bei der LPNMR 2022 publiziert. Diese Arbeit wurde als zweitbeste Arbeit ausgezeichnet (honorable mention<sup>1</sup>) und die Autoren bekamen die Einladung dazu ein Journal Paper im Special Issue des Journals *Theory and Practice of Logic Programming* zu veröffentlichen [Fichte et al., 2024]

**AP 3(b): Projektion.** In diesem Arbeitspaket wird untersucht, wie eine gewöhnliche Menge von Atomen auf ein gültiges Answer Set abgebildet werden kann. Das ist vor allem interessant wenn sich die Menge von Atomen nicht komplett abbilden lässt. Die Idee dahinter ist, dass die Anwenderin oder der Anwender eine Vorstellung hat, wie die Lösung zu dem in ASP modellierten Problem aussehen soll, diese jedoch nicht vollständig erfüllt werden kann. Ziel ist es, diese Menge von Atomen so optimal wie möglich zu erfüllen und somit das Answer Set zu finden, dass der Menge am ähnlichsten ist. Hier geht es wiederum nicht darum, das ASP Programm zu korrigieren, sondern in der Menge der gültigen Lösungen eine oder mehrere zu finden, die der Vorstellung entsprechen.

**Ergebnisse:** Projektion wurde in den Prototypen *fasb* (siehe [Fichte et al., 2022b]) teilweise mit integriert. Gegen Ende des Projekts begann eine Masterarbeit die sich intensiv mit dem Thema Projektion befasst, Ergebnisse dazu werden im Jahr 2025 erwartet.

## Arbeitspaket 4 – Implementierung

Die Implementierung eines Prototypen, der die entwickelten Navigationsstrategien vereint und auf effiziente Art und Weise umsetzt, ist eine zentrale Aufgabe des Projekts. Dabei fließen die Erkenntnisse aus den Analysen von **AP 1**, **AP 2** und **AP 3** ein. Der entwickelte Prototyp wird speziell in Hinsicht auf die Anforderungen für die beiden Anwendungsfälle in **AP 5** analysiert, und in **AP 6** evaluiert. Eine Unterscheidung zwischen Offline- und Online-Methoden wurde bereits in [Eiter et al., 2009, 2013] getroffen. Dabei versteht man unter *Online*-Methode, dass in den Lösungsprozess des ASP-Solvers selbst eingegriffen wird, während bei *Offline*-Methoden entweder der Input (das ASP Programm) verändert oder der Output (die Lösungen) im Nachhinein verarbeitet wird. Soweit möglich sollen die in **AP 2** und **AP 3** beschriebenen Navigationsstrategien sowohl mit Offline- als auch mit Online-Methoden implementiert werden. Die Evaluation in **AP 6** wird dann ergeben, welche Methode jeweils

<sup>1</sup><https://iccl.inf.tu-dresden.de/web/News78>

effizienter ist.

**AP 4(a): Offline-Methoden.** Unter Offline-Methoden verstehen wir hier, dass bei der Umsetzung der Navigationsmethode nicht in den Lösungsprozess des ASP-Solvers eingegriffen wird. Vielmehr konzentriert man sich hier auf die Vor- bzw. Nachbearbeitung des Programms bzw. der Menge der Answer Sets. Zum Beispiel können beim Faceted Browsing einzelne Atome fixiert werden, indem entsprechende integrity constraints zum Input des Programms hinzugenommen werden. In der Umsetzung wird besonderes Augenmerk auf die Resultate von **AP 1(b)** gelegt, um Algorithmen zu entwerfen, die passend zu den Komplexitätsresultaten der Navigationsstrategien sind. Obwohl hier nicht in den ASP Solver eingegriffen wird, ist es durchaus das Ziel, die Umsetzung der Methode von der eigentlichen Modellierung des Problems (insbesondere der Kodierung in ASP) unabhängig zu gestalten. Des Weiteren ist auch der Einsatz von fortgeschrittenen Techniken wie Multi-Shot ASP-Solving [Gebser et al., 2019] und Präferenzen wie in `asprin` [Brewka et al., 2015] denkbar.

**Ergebnisse:** Es hat sich herausgestellt, dass die Offline-Methoden deutlich einfacher umzusetzen sind, da sie in bestehende Solver integriert werden konnten, und dadurch auch für Anwender:Innen leichter zu verwenden sind. Zum Beispiel müssen dadurch nicht noch mal bestehende Solver neu kompiliert und installiert werden, was von Anwenderseite ein große Hürde für die Benutzung darstellen kann. Alle entwickelten Methoden wurden prototypisch implementiert und in den folgenden Publikationen veröffentlicht Fichte et al. [2022a,b], Böhl and Gaggl [2022], Gaggl et al. [2022], Böhl et al. [2023], Fichte et al. [2024], Böhl et al. [2024] Die Prototypen sind hier erhältlich:

- `fasb` <https://github.com/drwadu/fasb>
- `web-tool` von `fasb` <https://drwadu.github.io/web-fasb.github.io/>
- `iascar` <https://github.com/drwadu/iascar>
- `tunas` <https://doi.org/10.5281/zenodo.6762554>
- `snake` <https://github.com/elbo4u/asp-snake-ms>

**AP 4(b): Online-Methoden.** In diesem Arbeitspaket wird in den Lösungsprozess des prominenten open source ASP Solvers `clingo` eingegriffen, um die Navigationsstrategien zu implementieren. Der Lösungsprozess von `clingo` basiert auf der sogenannten *Conflict Driven Nogood Learning* Methode, dabei werden in der Suche nach Answer Sets entstandene Konflikte aufgelöst und in zusätzliche Informationen umgewandelt (gelernt), die dem Lösungsprozess zugeführt werden. Durch Heuristiken wird gesteuert welche nichtdeterministischen Entscheidungen in der Suche getroffen werden. In diesem Arbeitspaket wird in den Lösungsprozess eingegriffen, indem die in **AP 2** und **AP 3** entwickelten Methoden einerseits in Form von zusätzlichen Informationen eingebunden werden, andererseits werden darauf basierende neue Heuristiken definiert und implementiert. Es wird erwartet, dass man dadurch schneller und mit geringerem Speicherplatzbedarf zum Ergebnis kommt.

**Ergebnisse:** Die meisten Methoden wurden als Offline-Methoden implementiert (siehe Erklärung oben). In [Böhl et al., 2024] wurde aber auch eine Variante von Multi-Shot Solving untersucht, die mehr in die Funktionsweise vom Solver eingreift.

**AP 4(c): Graphische Darstellung.** Die graphische Darstellung und Aufbereitung von umfangreichen ASP-Lösungsmengen stellt eine besondere Herausforderung dar. Kleinere Lösungsräume können noch durchaus gut visualisiert werden, aber selbst bei der Darstellung von graph-basierten Problemen wie in Abstrakter Argumentation ist eine Visualisierung selbst von einzelnen Lösungen sehr großer Graphen nicht mehr übersichtlich realisierbar. Ziel dieses Arbeitspaketes ist, alternative Methoden zur graphischen Darstellung des Lösungsraumes der beiden Anwendungsfälle in **AP 5** zu erforschen. Zu **AP 5(a)**: AFs werden typischer Weise als Graphen dargestellt, und die Lösungen zu den einzelnen Semantiken sind durch Mengen von so genannten *Argumenten* gegeben, die meistens in dem Graphen (z.B. durch Färbung) hervorgehoben werden. Wie man sich gut vorstellen kann, ist ab einer gewissen Größe des Problems, sowohl der Graph selbst schwer übersichtlich darzustellen, als auch eine große Menge an Lösungen. Mithilfe von *Clustering* und *Zooming Methoden* werden alternative Visualisierungen realisiert. Beim Clustering wird analysiert in wie weit es Zusammenhänge von Answer Sets gibt, und wie diese visuell dargestellt werden können. Durch *Dimensional Reduction* wird der mehrdimensionale Answer Set Raum auf zweidimensionale und dreidimensionale Ebene reduziert. Answer Sets die sich dann nahe sind, werden zu Clustern zusammengefasst. Weiters werden Korrelationen zwischen Answer Sets analysiert und visuell aufbereitet.

Zooming Methoden ermöglichen es den Lösungsraum einerseits übersichtlich zusammenzufassen und gleichzeitig Details von besonders relevanten Bereichen höher aufzulösen.

**Ergebnisse:** Eine graphisch unterstützte Navigation durch Lösungsräume wurde in der Kooperation mit Prof. Raimund Dachzelt und seiner Gruppe in [Dachzelt et al., 2022] entwickelt. Anhand des Anwendungsfalles der abstrakten Argumentation wurde das Tool `nexas` entwickelt, das es erlaubt die Lösungsräume für zwei Semantiken zu einem Inputframework zu analysieren. Dabei wurde besonderes Augenmerk auf eine intuitive Darstellung und Visualisierung gelegt. Der Artikel und das Tool wurden auf der COMMA 2022 publiziert und präsentiert. Im Backend wurden auch faceted-navigation Methoden von **AP 3(a)** integriert. Das web-basierte Tool `nexas` ist hier verfügbar: <https://imld.de/en/research/research-projects/nexas/>

## Arbeitspaket 5 – Anwendungsfälle

Um die oben genannten Navigationsstrategien in realistischen Szenarien zu erproben und zu evaluieren, wurden die folgenden beiden Anwendungsfälle herangezogen.

**AP 5(a): Argumentation** Abstract Argumentation Frameworks (AFs) wurden durch Dung eingeführt [Dung, 1995] um Konflikte in nicht-monotone Formalismen zu modellieren und aufzulösen. Das Forschungsgebiet gewann in den letzten Jahren immer mehr Aufmerksamkeit, begründet durch sowohl die Einfachheit des Formalismus, als auch die hohe Komplexität der lösbaren Probleme. Mittlerweile existieren eine Vielzahl von Solvern, um die zugrunde liegenden Inferenzprobleme zu lösen, welche sich alle zwei Jahre bei der Internationalen Competition for Computational Models of Argumentation (ICMA) messen. ASPARTIX [Egly et al., 2010] war das erste System, das alle prominenten Semantiken von AFs unterstützte. Das ASP-basierte System zählt immer noch zu den Referenzsystemen und konnte auch bei

der letzten ICCMA 2019 [Bistarelli et al., 2018] einige führende Plätze erreichen.<sup>2</sup>

AFs eignen sich besonders gut als Anwendungsfall für NAVAS, da einerseits durch das System ASPARTIX eine umfassende Sammlung an ASP-Encodings zu allen prominenten Semantiken existieren, als auch der Lösungsraum für einige Semantiken enorm groß werden kann [Gaggl et al., 2020]. Die Erweiterung des Systems ASPARTIX um Navigationsstrategien wäre von großer Relevanz für die Argumentation Community. Dies würde es ermöglichen, einen genaueren Einblick in den Lösungsraum von AFs zu erlangen, wodurch weitere Rückschlüsse über das Verhalten der Semantiken gewonnen werden können.

**Ergebnisse:** Fast alle Methoden wurden auch an Argumentation Instanzen evaluiert. Dabei hat sich gezeigt, dass je nach Abhängigkeit der Schwere des zugrundeliegenden Problems, die Ansätze auch gut skalieren. Detaillierte Ergebnisse sind in den Publikationen zu finden [Fichte et al., 2022a,b, Böhl and Gaggl, 2022, Gaggl et al., 2022, Böhl et al., 2023, Fichte et al., 2024, Böhl et al., 2024]. Zusätzlich gibt es noch eine Liste an Arbeiten, die sich intensiver mit den Anwendungsfall Argumentation befasst haben [Diller et al., 2021, Baier et al., 2021, Käfer et al., 2022, Ellmauthaler et al., 2022a,b, Diller et al., 2022, Dachsel et al., 2022]

**AP 5(b): Konfiguration** ASP ist besonders gut geeignet, um kombinatorische Suchprobleme zu modellieren und zu lösen. Darunter ist Konfiguration ein besonders passender Anwendungsfall. Die modulare und deklarative Methodologie von ASP ermöglicht eine optimale Entwicklung von Programmen zur Lösung von Konfigurationsproblemen.

Für NAVAS sind Konfigurationsprobleme von großer Bedeutung, da der Lösungsraum in praktischen Anwendungen schnell Größen erreicht, die mit herkömmlichen Postprocessing-Ansätzen nicht mehr bewältigt werden können. Typisch für diesen Anwendungsfall ist auch, dass sich viele Lösungen sehr ähnlich sind. So kann zum Beispiel durch Elimination von inhaltlich redundanten, aber strukturell unterschiedlichen Lösungen, die eigentlich interessante Lösungsmenge deutlich kompakter dargestellt werden.

In diesem Arbeitspaket wurden die einzelnen Navigationsstrategien bezüglich ihrer Effizienz und Anwendbarkeit auf ASP Encodings zu einem möglichst generischen Konfigurationsproblem untersucht werden.

**Ergebnisse:** Die meisten Methoden wurden anhand des PC Konfigurators [Gorczyca, 2020] evaluiert. Damit können Konfigurationsinstanzen erstellt werden, die so groß sind, dass Enumeration nicht mehr möglich ist. In der jeweiligen Evaluierung konnten wir zeigen, dass die entwickelten Methoden auch für diese Art von Problemen funktionieren.

## Arbeitspaket 6 – Evaluation

In diesem Arbeitspaket wird der in **AP 4** entwickelte Prototyp evaluiert. Gegenstand der Evaluation ist in erster Linie die Performanz der Navigationsstrategien. Hierzu werden die einzelnen Offline- und Online-Methoden bezüglich ihrer Skalierbarkeit miteinander verglichen. Dazu bietet das Zentrum für Informationsdienste und Hochleistungsrechnen (ZIH) der TU Dresden hervorragende HPC Infrastruktur, die bereits in vorangegangenen Projekten und Wettbewerben genutzt wurde [Gaggl et al., 2020, 2015a].

<sup>2</sup><https://www.iccma2019.dmi.unipg.it/results/results-main.html>

Weiters dient die Evaluation der Qualitätskontrolle um mögliche Fehler in der Implementierung zu korrigieren. Dazu wird eine eigene Testumgebung entwickelt, die es ermöglicht, die Tests so weit wie möglich zu automatisieren. Eine frühzeitige und regelmäßige Evaluation des Prototypen ist entscheidend für den Erfolg des Projektes, denn dadurch wird die Performanz und die Korrektheit des Prototyps sichergestellt.

Für die Evaluierung werden Benchmarks der beiden Anwendungsfälle in **AP 5** herangezogen. Die Ergebnisse der Evaluation fließen dann wieder in die Weiterentwicklung und Verbesserung in **AP 4** ein.

**Ergebnisse:** Alle Methoden wurden hinsichtlich ihrer Performance und Umsetzbarkeit evaluiert, die Ergebnisse finden sich in den jeweiligen Publikationen.

## 4 Verwertungsplan

### 4.1 Wirtschaftliche Erfolge/Aussichten

Bei dem Projekt handelte es sich um anspruchsvolle Grundlagenforschung mit Anwendungsperspektive. Ziel der Forschung war es, die Grundlagen für einen breiteren Einsatz von ASP Technologie zu legen. Bevor diese jedoch zum Einsatz kommen kann, war es notwendig, die Möglichkeiten und Grenzen der Navigation im Lösungsraum von ASP Programmen zu erforschen. Wirtschaftliche Erfolgsaussichten sehen wir noch immer in der Zukunft, unter anderem in den Bereichen Planung und Produktkonfiguration.

### 4.2 Wissenschaftliche und technische Erfolge

Im Zuge des Projekts entstanden 24 Publikationen, darunter 4 Journalpublikationen, 14 Konferenzpublikationen, 3 Technische Berichte, 3 herausgegebene Tagungsbände, 6 eingeladene Vorträge, und eine Systembeschreibung durch die Teilnahme an dem ICCMA 2023 Wettbewerb (siehe [https://iccl.inf.tu-dresden.de/web/NAVAS/Publication\\_report/en](https://iccl.inf.tu-dresden.de/web/NAVAS/Publication_report/en)). Es gab eine abgeschlossene Dissertation (Verteidigung 14.5.2025), 3 abgeschlossene Master Arbeiten und eine abgeschlossene Bachelor Arbeit (siehe [https://iccl.inf.tu-dresden.de/web/Logische\\_Programmierung\\_und\\_Argumentation/Abgeschlossene\\_Studienarbeiten](https://iccl.inf.tu-dresden.de/web/Logische_Programmierung_und_Argumentation/Abgeschlossene_Studienarbeiten)).

PI Gaggl hat 2023 die internationale JELIA Konferenz in Dresden als General Chair organisiert (siehe <https://jelia2023.inf.tu-dresden.de/>).

PI Gaggl war bei der ICLP 2023 als Sprecherin eingeladen, musste jedoch kurzfristig krankheitsbedingt absagen (siehe <https://iclp2023.imperial.ac.uk/speakers>). Bei der NMR 2024 war PI Gaggl als Sprecherin eingeladen und hat die Fortschritte des Projekts präsentiert (siehe <https://nmr.krportal.org/2024/>).

Auf Outreach-Aktivitäten der TU Dresden (OUTPUT) wurde das ASP-basierte System snake von Elisa Böhl präsentiert [Böhl et al., 2024] (siehe <https://output-dd.de/blog/project-post/snakes-logic-is-everywhere/>).

### 4.3 Wissenschaftliche und wirtschaftliche Anschlussfähigkeit

Die erzielten Ergebnisse bekamen schon einiges an Anerkennung im wissenschaftlichen Umfeld. Einerseits durch Zitationen der veröffentlichten Arbeiten, andererseits durch Einla-

dungen die Ergebnisse bei internationalen Konferenzen, Workshops und Seminaren vorzustellen (siehe Abschnitt 4.2). Besonders die Arbeit zu **AP 3(a)** zeigt viel Potential für Anschlussarbeiten. Die Idee durch Lösungsräume zu navigieren beschränkt sich nicht nur auf ASP, sondern lässt sich auch verallgemeinern und hat viel Potential in anderen Bereichen wie zum Beispiel Planning.

Weitere Vorhaben die erzielten Ergebnisse auszubauen und zu vertiefen sind in Planung und führen womöglich in Zukunft zu weiteren Projektanträgen und Publikationen.

Alle entwickelten Tools sind offen zugänglich und können auch von einer breiten Nutzerschicht verwendet werden. Aktuell gibt es keine konkreten Industriepartner, jedoch wäre eine industrielle Nutzung der Methoden möglich und würde von PI Gaggl unterstützt werden.

#### **4.4 Bekannt gewordene Fortschritte auf dem Gebiet des Vorhabens bei anderen Stellen**

Das Vorhaben beinhaltete eine neue Idee, die nicht zur selben Zeit an anderen Stellen entwickelt wurde.

## **5 Zusammenarbeit mit Dritten**

Der regelmäßige Austausch mit den Kooperationspartnern ist wegweisend, da dadurch auch deren neueste Entwicklungen und Erkenntnisse berücksichtigt werden können. Aufgrund der Pandemie und der daraus resultierenden Einschränkungen waren keine regelmäßigen Workshops mit den Kooperationspartnern zu Beginn des Projektes möglich. Es gab einen Workshop 23.-24.05.2022 in Wien (siehe [https://iccl.inf.tu-dresden.de/web/Sarah\\_Alice\\_Gaggl/NAVAS\\_Workshop-Wien](https://iccl.inf.tu-dresden.de/web/Sarah_Alice_Gaggl/NAVAS_Workshop-Wien)), in dem es regen Austausch zu den eigenen und den Arbeiten des Kooperationspartners Thomas Eiter und der Wiener Kollegen gab. Mit den anderen externen Kooperationspartnern gab es virtuelle Treffen und Seminare.

**Raimund Dachsel, TU Dresden** Experte im Bereich Human Computer Interaction; Kooperation in **AP 4(c)**.

**Thomas Eiter, TU Wien** Führender Experte für theoretische Aspekte von ASP; Kooperation in **AP 1**.

**Wolfgang Faber, Universität Klagenfurt** Experte in Implementierung und praktischem Einsatz von ASP, Mitgründer von dlvsystems; Kooperation in **AP 4** und **AP 5**.

**Martin Gebser, Universität Klagenfurt** Experte in Implementierung und Einsatz von ASP Solvern. Kooperation in **AP 1** und **AP 4**.

**Sebastian Rudolph, TU Dresden** Expertise in praktischer Anwendung von ASP für Konfigurationsprobleme; Kooperation in **AP 5**.

**Torsten Schaub, Universität Potsdam** Führender Experte in der Implementierung effizienter ASP-Solver; Kooperation in **AP 4**, insbesondere **4(b)**.

## 6 Notwendigkeit und Angemessenheit der geleisteten Projektarbeiten

Für das hier vorgestellte Forschungsprojekt gab es keine alternative Finanzierung: Es handelte sich um ein risikoreiches, hoch innovatives Projekt mit hohem wissenschaftlichen und gesellschaftlichen Potenzial, weshalb eine öffentliche Förderung notwendig war. Seine Ausrichtung auf Grundlagenforschung schließt eine unmittelbare Industriefinanzierung oder eine Förderung im Rahmen einer der eher anwendungsorientierten EU-Förderlinien aus.

Projektlaufzeit war von 1.10.2020 bis 31.10.2024, dabei gab es eine kostenneutrale Verlängerung um einen Monat, weil unter anderem die Arbeit [Böhl et al., 2024] auf der ICLP 2024 präsentiert wurde. Um das Vorhaben umzusetzen wurden neben der Projektleiterin (49 Monate) eine Doktorandin (48 Monate) und ein Doktorand (46,5 Monate) beschäftigt. Der kürzere Zeitraum für den Doktoranden ergab sich durch einen späteren Start weil die Ausschreibung, Bewerbersichtung usw. erst eine spätere Einstellung erlaubt hat. Der Doktorand Dominik Rusovac hat Anfang Januar 2025 seine Dissertation eingereicht, die Verteidigung ist für den 14.5.2025 geplant. Frau Elisa Böhl konnte durch erhebliche Einschränkungen in der Pandemie durch familiäre Verpflichtungen die Dissertation noch nicht abschließen, dies soll im Nachgang an das Projekt erfolgen. Zusätzlich wurde ein WHK für 6 Monate angestellt, der bei der Implementierungen und Evaluierungen unterstützend eingesetzt wurde.

### 6.1 Wichtigste Positionen des zahlenmäßigen Nachweises

Position	Ausgaben
Beschäftigte E12-E15	914.306,00
Beschäftigungsentgelte	26.887,02
Dienstreisen	18.842,52
Gegenstände	2.140,08
Gesamtausgaben	962.831,87
Zuwendung zuzügl. Projektpauschale	1.155.398,25

## 7 Publikationen des Projekts

### 7.1 Artikel in Fachzeitschriften

- [j4] Elisa Böhl, Stefan Ellmauthaler, Sarah Alice Gaggl. Winning Snake: Design Choices in Multi-Shot ASP. In *Proceedings of the 40th International Conference on Logic Programming (ICLP 2024)*, Theory and Practice of Logic Programming, 24(4):772-789, July 2024.
- [j3] Johannes Klaus Fichte, Sarah Alice Gaggl, Markus Hecher, Dominik Rusovac. IAS-CAR: Incremental Answer Set Counting by Anytime Refinement. In *Theory and Practice of Logic Programming*, 1-28, February 2024.
- [j2] Nikolai Käfer, Christel Baier, Martin Diller, Clemens Dubslaff, Sarah Alice Gaggl, Holger Hermanns. Admissibility in Probabilistic Argumentation. In *Journal of Artificial Intelligence Research* 74 (2022).

- [j1] Sarah Alice Gaggl, Sebastian Rudolph, Hannes Strass. On the Decomposition of Abstract Dialectical Frameworks and the Complexity of Naive-based Semantics. In *Journal of Artificial Intelligence Research*, 70: 1–64 (2021).

## 7.2 Artikel in Tagungsbänden

- [c14]Dominik Rusovac, Markus Hecher, Martin Gebser, Sarah Alice Gaggl, Johannes K. Fichte. Navigating and Querying Answer Sets: How Hard Is It Really and Why? In *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning (KR 2024)*, 642–653, November 2024. IJCAI Organization.
- [c13]Elisa Böhl, Sarah Alice Gaggl, Dominik Rusovac. Representative Answer Sets: Collecting Something of Everything. In Kobi Gal, Ann Nowé, Grzegorz J. Nalepa, Roy Fairstein, Roxana Radulescu, eds., *Proceedings of the 26th European Conference on Artificial Intelligence (ECAI 2023)*, 271–278, September 2023. IOS Press.
- [c12]Martin Diller, Sarah Alice Gaggl, Piotr Gorczyca. Strategies in Flexible Dispute Derivations for Assumption-Based Argumentation. In Sarah Alice Gaggl, Jean-Guy Mailly, Matthias Thimm, Johannes Peter Wallner, eds., *Proceedings of the 4th International Workshop on Systems and Algorithms for Formal Argumentation (SAFA 2022)*, volume 3236, 59-72, October 2022. CEUR-WS.
- [c11]Raimund Dachzelt, Sarah Alice Gaggl, Markus Krötzsch, Julián Méndez, Dominik Rusovac, Mei Yang. NEXAS: A Visual Tool for Navigating and Exploring Argumentation Solution Spaces. In Francesca Toni, eds., *Proceedings of the 9th International Conference on Computational Models of Argument (COMMA 2022)*, volume 220146 of FAIA, 116-127, September 2022. IOS Press.
- [c10]Stefan Ellmauthaler, Sarah Alice Gaggl, Dominik Rusovac, Johannes Peter Wallner. ADF - BDD : An ADF Solver Based on Binary Decision Diagrams. In Francesca Toni, eds., *Proceedings of the 9th International Conference on Computational Models of Argument (COMMA 2022)*, volume 220146 of FAIA, 355-356, September 2022. IOS Press.
- [c9] Elisa Böhl, Sarah Alice Gaggl. Tunas - Fishing for Diverse Answer Sets: A Multi-Shot Trade up Strategy. In Georg Gottlob, Daniela Inclezan, Marco Maratea, eds., *Proceedings of the 16th International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR 2022)*, volume 13416 of Lecture Notes in Computer Science, 89-102, August 2022. Springer International Publishing.
- [c8] Johannes Klaus Fichte, Sarah Alice Gaggl, Markus Hecher, Dominik Rusovac. IAS-CAR: Incremental Answer Set Counting by Anytime Refinement. In Georg Gottlob, Daniela Inclezan, Marco Maratea, eds., *Proceedings of the 16th International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR 2022)*, volume 13416 of Lecture Notes in Computer Science, 217-230, August 2022. Springer International Publishing; honorable mention (runnerup for best paper award).

- [c7] Ridhwan Dewoprabowo, Johannes Klaus Fichte, Piotr Gorczyca, Markus Hecher. A Practical Account into Counting Dung’s Extensions by Dynamic Programming In Georg Gottlob, Daniela Inclezan, Marco Maratea, eds., *Proceedings of the 16th International Conference on Logic Programming and Non-monotonic Reasoning (LP-NMR 2022)*, volume 13416 of Lecture Notes in Computer Science, 387-400, 2022. Springer
- [c6] Stefan Ellmauthaler, Sarah Alice Gaggl, Dominik Rusovac, Johannes Peter Wallner. Representing Abstract Dialectical Frameworks with Binary Decision Diagrams. In Georg Gottlob, Daniela Inclezan, Marco Maratea, eds., *Proceedings of the 16th International Conference on Logic Programming and Non-monotonic Reasoning (LP-NMR 2022)*, volume 13416 of Lecture Notes in Computer Science, 177-198, 2022. Springer International Publishing.
- [c5] Johannes Klaus Fichte, Sarah Alice Gaggl, Dominik Rusovac. Rushing and Strolling among Answer Sets – Navigation Made Easy (Extended Abstract). In *Technical Communication Proceeding of the 38th International Conference on Logic Programming (ICLP 2022)*, to appear.
- [c4] Sarah Alice Gaggl, Philipp Hanisch, Markus Krötzsch. Simulating Sets in Answer Set Programming. In Luc De Raedt, eds., *Proceedings of the 31st International Joint Conference on Artificial Intelligence and the 25th European Conference on Artificial Intelligence (IJCAI-ECAI 2022)*, 2634–2640, 2022. ijcai.org, 2022.
- [c3] Johannes Klaus Fichte, Sarah Alice Gaggl, Dominik Rusovac. Rushing and Strolling among Answer Sets - Navigation Made Easy. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI 2022)*, volume 36 of 5, 5651–5659, 2022.
- [c2] Martin Diller, Sarah Alice Gaggl, Piotr Gorczyca. Flexible Dispute Derivations with Forward and Backward Arguments for Assumption-Based Argumentation. In Pietro Baroni, Christoph Benzmüller, Yi N. Wáng, eds., *Proceedings of the 4th International Conference on Logic and Argumentation (CLAR 2021)*, volume 13040 of Lecture Notes in Computer Science, 147–168, Springer, 2021.
- [c1] Christel Baier, Martin Diller, Clemens Dubsclaff, Sarah Alice Gaggl, Holger Hermanns, Nikolai Käfer. Admissibility in Probabilistic Argumentation. In Meghyn Bienvenu, Gerhard Lakemeyer, Esra Erdem, eds., *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning (KR 2021)*, 87–98, 2021.

### 7.3 Herausgegebene Tagungsbände

- [p3] Sarah Alice Gaggl, Maria Vanina Martinez, Magdalena Ortiz. Proceedings of the 18th European Conference on Logics in Artificial Intelligence (JELIA 2023), Dresden, Germany, September 20-22, 2023, Volume 14281 of Lecture Notes in Computer Science, September 2023. Springer.
- [p2] Enrico Pontelli, Stefania Costantini, Carmine Dodaro, Sarah Alice Gaggl, Roberta Calegari, Artur D’Avila Garcez, Francesco Fabiano, Francesco Fabiano, Alessandra Russo, Francesca Toni. Proceedings 39th International Conference on Logic Programming (ICLP 2023). Volume 385, September 2023. Open Publishing Association.

[p1] Sarah Alice Gaggl, Jean-Guy Mailly, Matthias Thimm, Johannes Peter Wallner. Proceedings of the The Fourth International Workshop on Systems and Algorithms for Formal Argumentation (SAFA 2022). Volume 3236, October 2022. CEUR-WS.

#### **7.4 Technische Berichte**

- [t3] Elisa Böhl, Stefan Ellmauthaler, Sarah Alice Gaggl. Winning Snake: Design Choices in Multi-Shot ASP. Technical Report, arXiv.org, volume arXiv:2408.08150, August 2024. To appear in Theory and Practice of Logic Programming (TLP), Proceedings of ICLP 2024.
- [t2] Johannes Klaus Fichte, Sarah Alice Gaggl, Markus Hecher, Dominik Rusovac. IAS-CAR: Incremental Answer Set Counting by Anytime Refinement. Technical Report, arXiv.org, volume arXiv:2311.07233, November 2023. Under consideration in Theory and Practice of Logic Programming (TLP).
- [t1] Johannes Klaus Fichte, Sarah Alice Gaggl, Dominik Rusovac. Rushing and Strolling among Answer Sets - Navigation Made Easy. Technical Report, arXiv.org, volume CoRR abs/2112.07596, December 2021.

## Literatur

- Weronika T. Adrian, Mario Alviano, Francesco Calimeri, Bernardo Cuteri, Carmine Dodaro, Wolfgang Faber, Davide Fuscà, Nicola Leone, Marco Manna, Simona Perri, Francesco Ricca, Pierfrancesco Veltri, and Jessica Zangari. The ASP system DLV: advancements and applications. *KI*, 32(2-3):177–179, 2018. doi: 10.1007/s13218-018-0533-0. URL <https://doi.org/10.1007/s13218-018-0533-0>.
- Christian Al-Rabaa, Sebastian Rudolph, and Lukas Schweizer. Faceted answer-set navigation. In Christoph Benzmüller, Francesco Ricca, Xavier Parent, and Dumitru Roman, editors, *Proceedings of the Second International Joint Conference on Rules and Reasoning, (RuleML+RR2018), Luxembourg, September 18-21*, volume 11092 of *Lecture Notes in Computer Science*, pages 211–225. Springer, September 2018. doi: 10.1007/978-3-319-99906-7\_14.
- Thomas Ambroz, Günther Charwat, Andreas Jusits, Johannes Peter Wallner, and Stefan Woltran. Arvis: Visualizing relations between answer sets. In Pedro Cabalar and Tran Cao Son, editors, *Proceedings of the 12th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2013), Corunna, Spain, September 15-19*, volume 8148 of *Lecture Notes in Computer Science*, pages 73–78. Springer, 2013. doi: 10.1007/978-3-642-40564-8\_8. URL [https://doi.org/10.1007/978-3-642-40564-8\\_8](https://doi.org/10.1007/978-3-642-40564-8_8).
- Christian Anger, Kathrin Konczak, Thomas Linke, and Torsten Schaub. A glimpse of answer set programming. *KI*, 19(1):12, 2005. URL [http://www.kuenstliche-intelligenz.de/index.php?id=no-6423&tx\\_ki\\_pi1%5BshowUId%5D=1054&cHash=2e21401053](http://www.kuenstliche-intelligenz.de/index.php?id=no-6423&tx_ki_pi1%5BshowUId%5D=1054&cHash=2e21401053).
- Christel Baier, Martin Diller, Clemens Dubslaff, Sarah Alice Gaggl, Holger Hermanns, and Nikolai Käfer. Admissibility in probabilistic argumentation. In Meghyn Bienvenu, Gerhard Lakemeyer, and Esra Erdem, editors, *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning (KR 2021)*, pages 87–98, November 2021.
- Stefano Bistarelli, Lars Kotthoff, Francesco Santini, and Carlo Taticchi. Containerisation and dynamic frameworks in iccma’19. In Matthias Thimm, Federico Cerutti, and Mauro Vallati, editors, *Proceedings of the Second International Workshop on Systems and Algorithms for Formal Argumentation (SAFA 2018) co-located with the 7th International Conference on Computational Models of Argument (COMMA 2018), Warsaw, Poland, September 11, 2018.*, volume 2171 of *CEUR Workshop Proceedings*, pages 4–9. CEUR-WS.org, 2018. URL [http://ceur-ws.org/Vol-2171/paper\\_1.pdf](http://ceur-ws.org/Vol-2171/paper_1.pdf).
- Elisa Böhl and Sarah Alice Gaggl. Tunas - fishing for diverse answer sets: A multi-shot trade up strategy. In Georg Gottlob, Daniela Incezan, and Marco Maratea, editors, *Proceedings of the 16th International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR 2022)*, volume 13416 of *Lecture Notes in Computer Science*, pages 89–102. Springer International Publishing, August 2022. doi: [https://doi.org/10.1007/978-3-031-15707-3\\_8](https://doi.org/10.1007/978-3-031-15707-3_8).
- Elisa Böhl, Sarah Alice Gaggl, and Dominik Rusovac. Representative answer sets: Collecting something of everything. In Kobi Gal, Ann Nowé, Grzegorz J. Nalepa, Roy Fairstein, and Roxana Radulescu, editors, *Proceedings of the 26th European Conference on Artificial Intelligence (ECAI 2023)*, pages 271–278. IOS Press, September 2023. doi: 10.3233/FAIA230280.
- Elisa Böhl, Stefan Ellmauthaler, and Sarah Alice Gaggl. Winning snake: Design choices in multi-shot ASP. *Theory and Practice of Logic Programming*, 24(4):772–789, July 2024. doi: [doi.org/10.1017/S1471068424000267](https://doi.org/10.1017/S1471068424000267).

- Piero Bonatti, Francesco Calimeri, Nicola Leone, and Francesco Ricca. *Answer Set Programming*, pages 159–182. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-14309-0. doi: 10.1007/978-3-642-14309-0\_8. URL [https://doi.org/10.1007/978-3-642-14309-0\\_8](https://doi.org/10.1007/978-3-642-14309-0_8).
- Gerhard Brewka, Thomas Eiter, and Mirosław Truszczyński. Answer set programming at a glance. *Communications of the ACM*, 54(12):92–103, 2011.
- Gerhard Brewka, James P. Delgrande, Javier Romero, and Torsten Schaub. asprin: Customizing answer set preferences without a headache. In *AAAI*, pages 1467–1474. AAAI Press, 2015.
- Paula-Andra Busoniu, Johannes Oetsch, Jörg Pührer, Peter Skocovsky, and Hans Tompits. Sealion: An eclipse-based IDE for answer-set programming with advanced debugging support. *TPLP*, 13(4-5):657–673, 2013. doi: 10.1017/S1471068413000410. URL <https://doi.org/10.1017/S1471068413000410>.
- Pedro Cabalar, Roland Kaminski, Philip Morkisch, and Torsten Schaub. telingo = ASP + time. In Marcello Balduccini, Yuliya Lierler, and Stefan Woltran, editors, *Proceedings of the 15th International Conference on Logic Programming and Nonmonotonic Reasoning, (LPNMR 2019), Philadelphia, PA, USA, June 3-7, 2019*, volume 11481 of *Lecture Notes in Computer Science*, pages 256–269. Springer, 2019. doi: 10.1007/978-3-030-20528-7\_19. URL [https://doi.org/10.1007/978-3-030-20528-7\\_19](https://doi.org/10.1007/978-3-030-20528-7_19).
- Francesco Calimeri, Wolfgang Faber, Martin Gebser, Giovambattista Ianni, Roland Kaminski, Thomas Krennwallner, Nicola Leone, Francesco Ricca, and Torsten Schaub. Asp-core-2: Input language format. Technical report, <https://www.mat.unical.it/aspcomp2013/files/ASP-CORE-2.01c.pdf>, 2012.
- Federico Cerutti, Sarah Alice Gaggl, Matthias Thimm, and Johannes Peter Wallner. Foundations of implementations for formal argumentation. *The IfCoLog Journal of Logics and their Applications; Special Issue Formal Argumentation*, 4:2623–2705, September 2017.
- Federico Cerutti, Sarah Alice Gaggl, Matthias Thimm, and Johannes Peter Wallner. Foundations of implementations for formal argumentation. In Pietro Baroni, Dov Gabbay, Massimiliano Giacomin, and Leendert van der Torre, editors, *Handbook on Formal Argumentation*, pages 688–767. College Publications, February 2018.
- Günther Charwat, Wolfgang Dvořák, Sarah Alice Gaggl, Johannes Peter Wallner, and Stefan Woltran. Methods for solving reasoning problems in abstract argumentation – a survey. *Artificial Intelligence Journal*, 220(0):28–63, March 2015. doi: 10.1016/j.artint.2014.11.008.
- Raimund Dachzelt, Sarah Alice Gaggl, Markus Krötzsch, Julián Méndez, Dominik Rusovac, and Mei Yang. Nexas: A visual tool for navigating and exploring argumentation solution spaces. In Francesca Toni, editor, *Proceedings of the 9th International Conference on Computational Models of Argument (COMMA 2022)*, volume 220146 of *FAIA*, pages 116–127. IOS Press, September 2022. doi: 10.3233/FAIA220146.
- Martin Diller, Sarah Alice Gaggl, and Piotr Gorczyca. Flexible dispute derivations with forward and backward arguments for assumption-based argumentation. In Pietro Baroni, Christoph Benzmüller, and Yi N. Wán, editors, *Proceedings of the 4th International Conference on Logic and Argumentation (CLAR 2021)*, volume 13040 of *LNCS*, pages 147–168. Springer, 2021.
- Martin Diller, Sarah Alice Gaggl, and Piotr Gorczyca. Strategies in flexible dispute derivations for assumption-based argumentation. In Sarah A. Gaggl, Jean-Guy Mailly, Matthias Thimm, and Johannes P. Wallner, editors, *Proceedings of the 4th International Workshop on Systems and Algorithms for Formal Argumentation (SAFA 2022)*, volume 3236, pages 59–72. CEUR-WS, October 2022.

- Carmine Dodaro, Philip Gasteiger, Kristian Reale, Francesco Ricca, and Konstantin Schekotihin. Debugging non-ground ASP programs: Technique and graphical tools. *TPLP*, 19(2):290–316, 2019. doi: 10.1017/S1471068418000492. URL <https://doi.org/10.1017/S1471068418000492>.
- Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–358, 1995. doi: 10.1016/0004-3702(94)00041-X. URL [https://doi.org/10.1016/0004-3702\(94\)00041-X](https://doi.org/10.1016/0004-3702(94)00041-X).
- Wolfgang Dvořák, Sarah Alice Gaggl, Johannes Peter Wallner, and Stefan Woltran. Making use of advances in answer-set programming for abstract argumentation systems. In Hans Tompits, Salvador Abreu, Johannes Oetsch, Jörg Pührer, Dietmar Seipel, Masanobu Umeda, and Armin Wolf, editors, *Proceedings of the 19th International Conference, (INAP 2011), and 25th Workshop on Logic Programming, (WLP 2011), Vienna, Austria, September 28-30, 2011, Revised Selected Papers*, volume 7773 of *Lecture Notes in Computer Science*, pages 114–133. Springer, 2013. doi: 10.1007/978-3-642-41524-1\_7.
- Wolfgang Dvořák, Sarah Alice Gaggl, Thomas Linsbichler, and Johannes Peter Wallner. Reduction-based approaches to implement modgil’s extended argumentation frameworks. In Eiter, Thomas, Strass, Hannes, Truszczyński, Mirosław, Woltran, and Stefan, editors, *Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation - Essays Dedicated to Gerhard Brewka on the Occasion of His 60th Birthday*, volume 9060 of *Lecture Notes in Computer Science*, pages 249–264. Springer International Publishing, 2015. doi: 10.1007/978-3-319-14726-0\_17.
- Uwe Egly, Sarah Alice Gaggl, and Stefan Woltran. Aspartix: Implementing argumentation frameworks using answer-set programming. In Maria Garcia de la Banda and Enrico Pontelli, editors, *Proceedings of the 24th International Conference on Logic Programming, (ICLP 2008)*, volume 5366 of *Lecture Notes in Computer Science*, pages 734–738. Springer, 2008. doi: 10.1007/978-3-540-89982-2.
- Uwe Egly, Sarah Alice Gaggl, and Stefan Woltran. Answer-set programming encodings for argumentation frameworks. *Argument & Computation*, 1(2):147–177, 2010. doi: 10.1080/19462166.2010.486479.
- Thomas Eiter, Esra Erdem, Halit Erdogan, and Michael Fink. Finding similar or diverse solutions in answer set programming. In Patricia M. Hill and David Scott Warren, editors, *Proceedings of the 25th International Conference on Logic Programming (ICLP 2009)*, volume 5649 of *Lecture Notes in Computer Science*, pages 342–356. Springer, 2009. doi: 10.1007/978-3-642-02846-5\_29. URL [https://doi.org/10.1007/978-3-642-02846-5\\_29](https://doi.org/10.1007/978-3-642-02846-5_29).
- Thomas Eiter, Esra Erdem, Halit Erdogan, and Michael Fink. Finding similar/diverse solutions in answer set programming. *TPLP*, 13(3):303–359, 2013. doi: 10.1017/S1471068411000548. URL <https://doi.org/10.1017/S1471068411000548>.
- Thomas Eiter, Gerhard Friedrich, Richard Taupe, and Antonius Weinzierl. Lazy grounding for dynamic configuration - efficient large-scale (re)configuration of cyber-physical systems with ASP. *KI*, 32(2-3):197–198, 2018. doi: 10.1007/s13218-018-0536-x. URL <https://doi.org/10.1007/s13218-018-0536-x>.
- Stefan Ellmauthaler, Sarah Alice Gaggl, Dominik Rusovac, and Johannes P. Wallner. Adf - BDD : An ADF solver based on binary decision diagrams. In Francesca Toni, editor, *Proceedings of the 9th International Conference on Computational Models of Argument (COMMA 2022)*, volume 220146 of *FAIA*, pages 355–356. IOS Press, September 2022a. doi: 10.3233/FAIA220170.

- Stefan Ellmauthaler, Sarah Alice Gaggl, Dominik Rusovac, and Johannes P. Wallner. Representing abstract dialectical frameworks with binary decision diagrams. In G. Gottlob, D Incezan, and M. Maratea, editors, *Proceedings of the 16th International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR 2022)*, volume 13416 of *Lecture Notes in Computer Science*, pages 177–198. Springer, 2022b. doi: 10.1007/978-3-031-15707-3\_14.
- Esra Erdem and Volkan Patoglu. Applications of ASP in robotics. *KI*, 32(2-3):143–149, 2018. doi: 10.1007/s13218-018-0544-x. URL <https://doi.org/10.1007/s13218-018-0544-x>.
- Esra Erdem, Michael Gelfond, and Nicola Leone. Applications of answer set programming. *AI Magazine*, 37(3):53–68, 2016. URL <http://www.aaai.org/ojs/index.php/aimagazine/article/view/2678>.
- Andreas A. Falkner, Gerhard Friedrich, Konstantin Schekotihin, Richard Taupe, and Erich Christian Teppan. Industrial applications of answer set programming. *KI*, 32(2-3):165–176, 2018. doi: 10.1007/s13218-018-0548-6. URL <https://doi.org/10.1007/s13218-018-0548-6>.
- Onofrio Febbraro, Kristian Reale, and Francesco Ricca. ASPIDE: integrated development environment for answer set programming. In James P. Delgrande and Wolfgang Faber, editors, *Proceedings of the 11th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2011), Vancouver, Canada, May 16-19*, volume 6645 of *Lecture Notes in Computer Science*, pages 317–330. Springer, 2011. doi: 10.1007/978-3-642-20895-9\_37. URL [https://doi.org/10.1007/978-3-642-20895-9\\_37](https://doi.org/10.1007/978-3-642-20895-9_37).
- Johannes Klaus Fichte, Sarah Alice Gaggl, Markus Hecher, and Dominik Rusovac. Iascar: Incremental answer set counting by anytime refinement. In Georg Gottlob, Daniela Incezan, and Marco Maratea, editors, *Proceedings of the 16th International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR 2022)*, volume 13416 of *Lecture Notes in Computer Science*, pages 217–230. Springer International Publishing, August 2022a. doi: [https://doi.org/10.1007/978-3-031-15707-3\\_17](https://doi.org/10.1007/978-3-031-15707-3_17).
- Johannes Klaus Fichte, Sarah Alice Gaggl, and Dominik Rusovac. Rushing and strolling among answer sets - navigation made easy. In *Proceedings of the 36th AAI Conference on Artificial Intelligence (AAAI 2022)*, volume 36 of 5, pages 5651–5659, June 2022b. doi: <https://doi.org/10.1609/aaai.v36i5.20506>.
- Johannes Klaus Fichte, Sarah Alice Gaggl, Markus Hecher, and Dominik Rusovac. Iascar: Incremental answer set counting by anytime refinement. *Theory and Practice of Logic Programming*, 24(3): 505–532, May 2024. doi: <https://doi.org/10.1017/S1471068424000036>.
- Sarah Alice Gaggl and Hannes Strass. Decomposing abstract dialectical frameworks. In Simon Parsons, Nir Oren, Chris Reed, and Federico Cerutti, editors, *Proceedings of the 5th International Conference on Computational Models of Argument (COMMA 2014)*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 281–292. IOS Press, 2014. doi: 10.3233/978-1-61499-436-7-281.
- Sarah Alice Gaggl and Stefan Woltran. The cf2 argumentation semantics revisited. *Journal of Logic and Computation*, 23(5):925–949, 2013. doi: 10.1093/logcom/exs011.
- Sarah Alice Gaggl, Norbert Manthey, Alessandro Ronca, Johannes Peter Wallner, and Stefan Woltran. Improved answer-set programming encodings for abstract argumentation. *Theory and Practice of Logic Programming (TPLP)*, 15(4-5):434–448, September 2015a.

- Sarah Alice Gaggl, Sebastian Rudolph, and Lukas Schweizer. Bound your models! how to make OWL an ASP modeling language. In Stefan Ellmauthaler and Claudia Schulz, editors, *International Workshop on User-Oriented Logic Programming (IULP)*, November 2015b.
- Sarah Alice Gaggl, Sebastian Rudolph, and Hannes Strass. On the computational complexity of naive-based semantics for abstract dialectical frameworks. In Qiang Yang and Michael Wooldridge, editors, *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI 2015)*, AAAI Press, pages 2985–2991. AAAI Press, July 2015c.
- Sarah Alice Gaggl, Thomas Linsbichler, Marco Maratea, and Stefan Woltran. Introducing the second international competition on computational models of argumentation. In Matthias Thimm, Federico Cerutti, Hannes Strass, and Mauro Vallati, editors, *Proceedings of the First International Workshop on Systems and Algorithms for Formal Argumentation (SAFA 2016)*, volume 1672, pages 4–9. CEUR-WS, September 2016a.
- Sarah Alice Gaggl, Sebastian Rudolph, and Lukas Schweizer. Fixed-domain reasoning for description logics. In Gal A. Kaminka, Maria Fox, Paolo Bouquet, Eyke Hüllermeier, Virginia Dignum, Frank Dignum, and Frank van Harmelen, editors, *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI 2016)*, volume 285 of *Frontiers in Artificial Intelligence and Applications*, pages 819 – 827. IOS Press, September 2016b.
- Sarah Alice Gaggl, Thomas Linsbichler, Marco Maratea, and Stefan Woltran. Summary report of the second international competition on computational models of argumentation. *AI Magazine*, 39(4): 77–79, December 2018.
- Sarah Alice Gaggl, Thomas Linsbichler, Marco Maratea, and Stefan Woltran. Design and results of the second international competition on computational models of argumentation. Technical report, February 2020.
- Sarah Alice Gaggl, Philipp Hanisch, and Markus Krötzsch. Simulating sets in answer set programming. In Luc De Raedt, editor, *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI 2022)*, pages 2634–2640. ijcai.org, 2022. doi: 10.24963/ijcai.2022/365.
- Philip Gasteiger, Carmine Dodaro, Benjamin Musitsch, Kristian Reale, Francesco Ricca, and Konstantin Schekotihin. An integrated graphical user interface for debugging answer set programs. *CoRR*, abs/1611.04969, 2016. URL <http://arxiv.org/abs/1611.04969>.
- Martin Gebser, Benjamin Kaufmann, Roland Kaminski, Max Ostrowski, Torsten Schaub, and Marius Thomas Schneider. Potassco: The potsdam answer set solving collection. *AI Commun.*, 24(2): 107–124, 2011. doi: 10.3233/AIC-2011-0491. URL <https://doi.org/10.3233/AIC-2011-0491>.
- Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. *Answer Set Solving in Practice*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2012. doi: 10.2200/S00457ED1V01Y201211AIM019. URL <https://doi.org/10.2200/S00457ED1V01Y201211AIM019>.
- Martin Gebser, Marco Maratea, and Francesco Ricca. The sixth answer set programming competition. *J. Artif. Intell. Res.*, 60:41–95, 2017. doi: 10.1613/jair.5373. URL <https://doi.org/10.1613/jair.5373>.
- Martin Gebser, Roland Kaminski, Benjamin Kaufmann, and Torsten Schaub. Multi-shot ASP solving with clingo. *TPLP*, 19(1):27–82, 2019. doi: 10.1017/S1471068418000054. URL <https://doi.org/10.1017/S1471068418000054>.

- Michael Gelfond and Vladimir Lifschitz. The stable model semantics for logic programming. In Robert A. Kowalski and Kenneth A. Bowen, editors, *Proceedings of the Fifth International Conference and Symposium of Logic Programming (ICLP/SLP 1988)*, pages 1070–1080. MIT Press, 1988. ISBN 0-262-61056-6.
- Michael Gelfond and Vladimir Lifschitz. Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9(3/4):365–386, 1991.
- Eray Gençay, Peter Schüller, and Esra Erdem. Applications of non-monotonic reasoning to automotive product configuration using answer set programming. *J. Intelligent Manufacturing*, 30(3):1407–1422, 2019. doi: 10.1007/s10845-017-1333-3. URL <https://doi.org/10.1007/s10845-017-1333-3>.
- Piotr Gorczyca. Configuration problem asp encoding generator, November 2020. URL <https://doi.org/10.5281/zenodo.5777217>.
- Nikolai Käfer, Christel Baier, Martin Diller, Clemens Dubsloff, Sarah Alice Gaggl, and Holger Hermanns. Admissibility in probabilistic argumentation. *Journal of Artificial Intelligence Research*, 74:957–1004, June 2022. doi: <https://doi.org/10.1613/jair.1.13530>.
- Muhammed Kerem Kahraman and Esra Erdem. Personalized course schedule planning using answer set programming. In José Júlio Alferes and Moa Johansson, editors, *Proceedings of the 21th International Symposium on Practical Aspects of Declarative Languages (PADL 2019)*, volume 11372 of *Lecture Notes in Computer Science*, pages 37–45. Springer, 2019. doi: 10.1007/978-3-030-05998-9\\_3. URL [https://doi.org/10.1007/978-3-030-05998-9\\_3](https://doi.org/10.1007/978-3-030-05998-9_3).
- Benjamin Kaufmann, Nicola Leone, Simona Perri, and Torsten Schaub. Grounding and solving in answer set programming. *AI Magazine*, 37(3):25–32, 2016. URL <http://www.aaai.org/ojs/index.php/aimagazine/article/view/2672>.
- Christian Kloimüller, Johannes Oetsch, Jörg Pührer, and Hans Tompits. Kara: A system for visualising and visual editing of interpretations for answer-set programs. In Hans Tompits, Salvador Abreu, Johannes Oetsch, Jörg Pührer, Dietmar Seipel, Masanobu Umeda, and Armin Wolf, editors, *19th International Conference on Applications of Declarative Programming and Knowledge Management (INAP 2011), and 25th Workshop on Logic Programming, (WLP 2011), Vienna, Austria, September 28-30, 2011, Revised Selected Papers*, volume 7773 of *Lecture Notes in Computer Science*, pages 325–344. Springer, 2011. doi: 10.1007/978-3-642-41524-1\\_20. URL [https://doi.org/10.1007/978-3-642-41524-1\\_20](https://doi.org/10.1007/978-3-642-41524-1_20).
- Nicola Leone, Gerald Pfeifer, Wolfgang Faber, Thomas Eiter, Georg Gottlob, Simona Perri, and Francesco Scarcello. The DLV system for knowledge representation and reasoning. *ACM Trans. Comput. Log.*, 7(3):499–562, 2006. doi: 10.1145/1149114.1149117. URL <https://doi.org/10.1145/1149114.1149117>.
- Johannes Oetsch, Jörg Pührer, and Hans Tompits. Stepwise debugging of answer-set programs. *TPLP*, 18(1):30–80, 2018. doi: 10.1017/S1471068417000217. URL <https://doi.org/10.1017/S1471068417000217>.
- Javier Romero, Torsten Schaub, and Philipp Wanko. Computing diverse optimal stable models. In Manuel Carro, Andy King, Neda Saeedloei, and Marina De Vos, editors, *Technical Communications of the 32nd International Conference on Logic Programming, ICLP 2016 TCs, October 16-21, 2016, New York City, USA*, volume 52 of *OASICS*, pages 3:1–3:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi: 10.4230/OASICS.ICLP.2016.3. URL <https://doi.org/10.4230/OASICS.ICLP.2016.3>.

- Dominik Rusovac, Markus Hecher, Martin Gebser, Sarah Alice Gaggl, and Johannes K. Fichte. Navigating and querying answer sets: How hard is it really and why? In Pierre Marquis, Magdalena Ortiz, and Maurice Pagnucco, editors, *Proceedings of the 21st International Conference on Principles of Knowledge Representation and Reasoning (KR 2024)*, page 642–653. IJCAI Organization, November 2024. doi: <https://doi.org/10.24963/kr.2024/60>.
- Kostyantyn M. Shchekotykhin. Interactive query-based debugging of ASP programs. In Blai Bonet and Sven Koenig, editors, *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, (AAAI 2015), Austin, Texas, USA.*, pages 1597–1603. AAAI Press, 2015. URL <http://www.aaai.org/ocs/index.php/AAAI/AAAI15/paper/view/9400>.
- Richard Taupe, Antonius Weinzierl, and Gerhard Friedrich. Degrees of laziness in grounding - effects of lazy-grounding strategies on ASP solving. In Marcello Balduccini, Yuliya Lierler, and Stefan Woltran, editors, *Proceedings of the 15th International Conference on Logic Programming and Non-monotonic Reasoning (LPNMR 2019), Philadelphia, PA, USA, June 3-7*, volume 11481 of *Lecture Notes in Computer Science*, pages 298–311. Springer, 2019. doi: 10.1007/978-3-030-20528-7\_22. URL [https://doi.org/10.1007/978-3-030-20528-7\\_22](https://doi.org/10.1007/978-3-030-20528-7_22).
- Matthias Thimm and Serena Villata. The first international competition on computational models of argumentation: Results and analysis. *Artif. Intell.*, 252:267–294, 2017. doi: 10.1016/j.artint.2017.08.006. URL <https://doi.org/10.1016/j.artint.2017.08.006>.
- Matthias Thimm, Serena Villata, Federico Cerutti, Nir Oren, Hannes Strass, and Mauro Vallati. Summary report of the first international competition on computational models of argumentation. *AI Magazine*, 37(1):102, 2016. URL <http://www.aaai.org/ojs/index.php/aimagazine/article/view/2640>.
- Marina De Vos, Doga Gizem Kisa, Johannes Oetsch, Jörg Pührer, and Hans Tompits. Annotating answer-set programs in Iana. *TPLP*, 12(4-5):619–637, 2012. doi: 10.1017/S147106841200021X. URL <https://doi.org/10.1017/S147106841200021X>.