

Computationale Aspekte der Abstrakten Argumentation*

Wolfgang Dvořák

Universität Wien, Fakultät für Informatik
wolfgang.dvorak@univie.ac.at

Abstract: Diese Arbeit studiert *Abstract Argumentation Frameworks*, einen der erfolgreichsten Formalismen in der Formalen Argumentation, einem Teilgebiet der Künstlichen Intelligenz. Diese Frameworks abstrahieren von den konkreten Argumenten und ihren Inhalten zu abstrakten Entitäten und einer Konfliktrelation zwischen diesen. Auf Basis der Konfliktrelation werden nun kohärente Mengen von Argumenten identifiziert. Die Literatur kennt eine Vielzahl an unterschiedlichen Kriterien, sogenannte Semantiken, um kohärente Mengen zu definieren. Die Berechnung dieser kohärenten Mengen ist eine Kernaufgabe von Argumentations-Systemen weshalb sich diese Arbeit der computationalen Analyse dieses Schrittes widmet. Konkret präsentieren wir eine klassische *Komplexitätsanalyse* der verschiedenen Semantiken und studieren mögliche *tractable fragments*, d.h. wir betrachten Abstract Argumentation Frameworks mit einer bestimmte Struktur und untersuchen, ob diese mit geringerem Berechnungsaufwand ausgewertet werden können. Die erzielten Resultate nutzen wir abschließend auch um effiziente *Übersetzbarkeit* zwischen den verschiedenen Semantiken für Abstract Argumentation Frameworks zu studieren. Solche Übersetzungen erlauben die Nutzung eines effizienten Argumentations-Systems für eine Semantik A auch für andere Semantiken, die in Semantik A übersetzt werden können. Diese Untersuchungen bilden gemeinsam die Grundlage für das Design effizienter Argumentations-Systeme, welche eine breite Auswahl an Semantiken bereitstellen sollen.

1 Motivation

Ausgehend von Phan Minh Dungs wegweisender Arbeit [Dun95] hat sich das Forschungsgebiet der Argumentation zu einem der aktivsten in der Künstlichen Intelligenz (KI) entwickelt. Alle großen KI Konferenzen listen heutzutage Argumentation als Thema und unzählige Artikel im Bereich der Argumentation wurden in angesehenen Zeitschriften publiziert. Informell beschrieben beschäftigt sich Argumentation mit der Konstruktion von Argumenten, dem Finden von Konflikten zwischen Argumenten, und der Auswahl von kohärenten Mengen von Argumenten. Das Ziel eines solchen Prozesses ist es üblicher Weise eine Form von Wissen abzuleiten und eine Entscheidung für eine gegebene Problemstellung zu treffen. Solche Argumentations-Szenarien können aus ganz unterschiedlichen Kontexten stammen. Um zwei Kontrastbeispiele zu geben: es kann sich einerseits um einen einzelnen Agenten handeln, der sein Wissen auswertet um eine Entscheidung

*Englischer Titel der Dissertation: Computational Aspects of Abstract Argumentation.

über die nächste Aktion zu treffen, aber andererseits auch um eine Gruppe von Agenten, die Argumente austauschen um über die Nutzung einer Ressource zu verhandeln.

Das Wesen von Argumenten ist, dass sie prinzipiell durch neue Argumente anfechtbar sind. Argumentation ist also eine Form des nicht-monotonen Schließens, d.h. eine Folgerung aus dem aktuellen Wissen kann durch neues Wissen (und damit neuen Argumenten) widerlegt werden. Formale Argumentation wurde zu anderen populären nicht-monotonen Formalismen in Beziehung gesetzt. Dabei kann man beobachten, dass sich nicht-monotone Formalismen als Argumentations-Prozess realisieren lassen, indem zuerst Argumente, die anfechtbaren Beweisen im ursprünglichen Formalismus entsprechen, gebildet werden, dann Konflikte zwischen diesen Beweisen eruiert und abschließend kohärente Mengen von Argumenten ermittelt werden. Konkrete Beispiele für solche Formalismen sind "Default Logic" [Dun95], "Defeasible Logic" [GMAB04] und "Answer Set Programming" [Dun95]. Neben der KI beschäftigen sich auch andere Wissenschaften mit verschiedenen Formen von Argumentation und dementsprechend gibt es zahlreiche interdisziplinäre Schnittstellen zur Philosophie, Linguistik und der Rechtswissenschaft [BCD07].

Der Kern jedes (computerunterstützten) Argumentations-Systems ist eine Komponente, die kohärente Mengen von Argumenten findet. Für eine praktische Realisierung ist es essentiell die computationalen Eigenschaften der damit verbunden Problemstellungen zu verstehen. Ein Aspekt hier ist die klassische *Komplexitätsanalyse*, die einerseits Schranken für das Verhalten von bestmöglichen Algorithmen gibt und andererseits ein Maß für die Ausdrucksstärke verschiedener Semantiken für Argumentation darstellt. Unter einer Semantik versteht man dabei einen Regelsatz, der festlegt welche Mengen von Argumenten kohärent sind. Diese Analyse zeigt, dass viele Problemstellungen NP-schwer sind und daher Algorithmen nicht für alle Instanzen effizient sein können (unter der Annahme $P \neq NP$).

Trotz dieser NP-Schwere gibt es zahlreiche Instanzen, die in Polynomialzeit gelöst werden können. Für das Design von Systemen, die jede Instanz so effizient wie möglich verarbeiten, ist es also unabdingbar Klassen solcher Instanzen, sogenannte "*tractable fragments*" zu identifizieren. Eine weitere Möglichkeit für eine differenzierte Komplexitätsanalyse kommt aus der *Parametrisierten Komplexitätstheorie* (siehe u.a. [Nie06]). Während klassische Komplexitätstheorie die Komplexität im Verhältnis zur Größe der Instanz misst, erlaubt es die parametrisierte Komplexitätstheorie die Komplexität in Bezug auf Problem-Parameter anzugeben. Für viele NP-schwere Probleme gibt es Algorithmen, deren Laufzeit zwar exponentiell mit dem Parameter wächst, aber nur polynomiell mit der Größe der Instanz. Betrachtet man nur Instanzen bei denen der Parameter beschränkt ist, kann man diese in Polynomialzeit lösen, man nennt das Problem dann "*fixed-parameter tractable*".

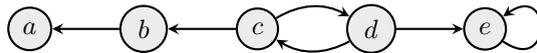
Ein weiterer Aspekt ist die Vielzahl an unterschiedlichen Semantiken, die es unrealistisch erscheinen lässt für jede ein effizientes System zu kreieren. Um dieses Problem zu handhaben kann man Semantiken ineinander übersetzen, d.h. Probleminstanzen so modifizieren, dass eine andere Semantik B das gleiche Resultat liefert wie die ursprüngliche Semantik A auf der ursprünglichen Instanz. So eine *Übersetzung von Semantiken* erlaubt es dann ein ausgefeiltes System für Semantik B auch für Semantik A zu verwenden.

In dieser Arbeit geben wir eine umfassende Analyse dieser Aspekte für Abstrakte Argumentation [Dun95], die den meisten Argumentations-Systemen zu Grunde liegt [BCD07].

2 Argumentation in der Künstlichen Intelligenz

In diesem Abschnitt stellen wir Argumentation in der KI in der hier gebotenen Kürze vor. Für eine ausführliche Einführung sei der interessierte Leser an den Überblicksartikel von Bench-Capon und Dunne [BCD07] verwiesen.

Abstrakte Argumentation Diese Arbeit konzentriert sich auf Abstrakte Argumentation, den zentralen Formalismus im Gebiet der Formalen Argumentation. Der Kern der Argumentation ist das "Auflösen" von Konflikten zwischen Argumenten durch das Identifizieren von in sich schlüssigen Mengen von Argumenten. Dieser Schritt kann durch sogenannte *Abstract Argumentation Frameworks (AFs)* [Dun95] formalisiert werden. So ein AF besteht einerseits aus abstrakten Objekten, die Argumente repräsentieren, und andererseits aus einer Menge von Attacken zwischen diesen Argumenten die Konflikte modellieren. Dabei geht eine Attacke immer von genau einem Argument aus und attackiert genau ein Argument. AFs kümmern sich aber weder um die Aussage oder innere Struktur eines Arguments noch um die Ursache der Konflikte zwischen Argumenten. Dieses Modell kann als gerichteter Graph interpretiert werden, wobei die Knoten des Graphen den Argumenten entsprechen und die gerichteten Kanten des Graphen den Attacken. Zum Beispiel kann das AF das aus den fünf Argumenten a, b, c, d, e besteht und die Attacken (b, a) , (c, b) , (c, d) , (d, c) , (d, e) , (e, e) enthält wie folgt als Graph visualisiert werden:



Abstrakte Argumentation basiert auf dieser Graph Struktur und identifiziert mittels dieser kohärente Mengen von Argumenten. Die Literatur kennt eine Vielzahl unterschiedlicher Kriterien, die kohärente Mengen erfüllen sollen, einen Satz solcher Kriterien nennen wir *Semantik* (einen guten Überblick liefert [BCG11]). Zwei grundlegende Konzepte für diese Semantiken sind die Konfliktfreiheit und die Verteidigung von Argumenten. Wir nennen eine Menge von Argumenten konfliktfrei, wenn es keine Attacke zwischen den Argumenten in der Menge gibt. Eine Menge M verteidigt ein Argument a , wenn jedes Argument b das a attackiert von einem Argument $c \in M$ attackiert wird. Konfliktfreie Mengen die alle ihre Argumente auch verteidigt werden "admissible" genannt und bilden die Grundlage für den Großteil der betrachteten Semantiken. Die verschiedenen Semantiken sind durch unterschiedliche Szenarien und Intuitionen begründet und haben daher auch unterschiedlichste Eigenschaften. Dementsprechend beschäftigt sich eine Vielzahl von Arbeiten mit den Eigenschaften, die "gute" Argumentations-Semantiken erfüllen sollen, Implikationen solcher Eigenschaften, und den Beziehungen zwischen unterschiedlichen Semantiken (siehe z.B. [CA07, BCG11]). Diese Arbeit ergänzt diese Bemühungen durch eine Analyse der computationalen Eigenschaften der Semantiken.

Der Argumentations-Prozess Wir betrachten Abstrakte Argumentation als den zentralen Formalismus von Argumentation, nichtsdestotrotz ist Abstrakte Argumentation immer in eine Form von Argumentations-Prozess eingebettet. Schon in Dung's Arbeit werden Abstract Argumentation Frameworks gemeinsam mit mehreren Methoden AFs aus Wissensbasen zu konstruieren, sogenannten Instanzierungen von AFs, vorgestellt.

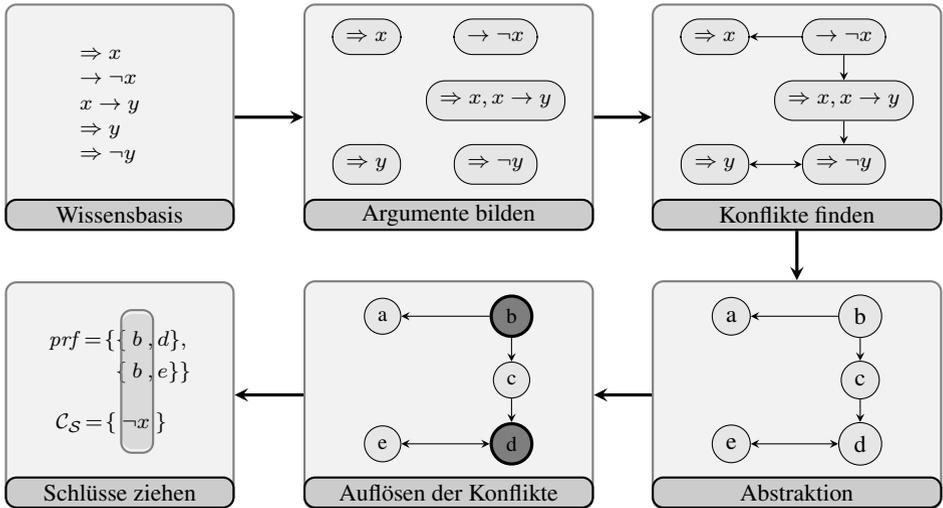


Abbildung 1: Eine Illustration des Argumentations-Prozesses für Defeasible Logic.

Im folgenden beschreiben wir das Konzept dieses Argumentations-Prozess (angelehnt an [CA07]). Dabei unterteilen wir diesen Prozess in sechs Schritte (siehe Abbildung 1): *Generieren einer Wissensbasis:* Eine Argumentation benötigt eine Form von Wissensbasis über die Problemstellung. Das kann eine Theorie in einer formalen Sprache sein oder einfach das Wissen eines Experten, der die Argumente konstruiert. *Konstruktion von Argumenten:* In einem ersten Schritt verwendet man die Wissensbasis um Argumente zu bilden. *Identifizieren von Konflikten:* Hier nutzt man die innere Struktur der Argumente um Konflikte zwischen den verschiedenen Argumenten zu identifizieren. *Abstraktion von der inneren Struktur der Argumente:* Das Argumentationsproblem wird auf ein Abstract Argumentation Framework reduziert und die innere Struktur der Argumente “vergessen”. *Finden der akzeptablen Mengen von Argumenten:* Auf dieser abstrakten Ebene berechnen wir kohärente Mengen von Argumenten. *Schlüsse ziehen:* Als abschließender Schritt wird auf Basis der kohärenten Mengen eine Entscheidung über einen Sachverhalt getroffen, indem man wieder die innere Struktur der akzeptierten Argumente betrachtet.

Computationale Problemstellungen Die zentrale Aufgabe eines Argumentations-Systems ist das Auffinden von kohärenten Mengen von Argumenten. Da Semantiken für solche kohärenten Mengen auf dem abstrakten Level arbeiten, sind eben Abstract Argumentation Frameworks das passende Modell um diese Aufgabe zu studieren. Will man Schlüsse über einen bestimmten Sachverhalt ziehen, ist man aber nicht immer direkt am Berechnen aller kohärenten Mengen interessiert, sondern viel mehr an einem konkreten Argument. Typische computationale Aufgaben sind (neben dem Berechnen von kohärenten Mengen) gutgläubiges (*Cred*) / skeptisches (*Skept*) Schließen, d.h. zu entscheiden ob ein spezielles Argument in mindestens einer / in allen kohärenten Mengen enthalten ist, das Verifizieren einer kohärenten Menge (*Ver*), d.h. zu entscheiden ob eine konkrete Menge kohärent ist, und zu Entscheiden ob eine (nicht-leere) kohärente Menge existiert

Tabelle 1: Komplexitätsklassifizierungen von Abstrakter Argumentation (C -c bedeutet vollständig für die Klasse C). Neue Resultate sind in Fettschrift hervorgehoben.

σ	$Cred_\sigma$	$Skept_\sigma$	$Ideal_\sigma$	Ver_σ	$Exists_\sigma^{-0}$
<i>cf</i>	in L	trivial	trivial	in L	in L
<i>naive</i>	in L	in L	P-c	in L	in L
<i>grd</i>	P-c	P-c	P-c	P-c	in L
<i>stb</i>	NP-c	coNP-c	D^P-c	in L	NP-c
<i>adm</i>	NP-c	trivial	trivial	in L	NP-c
<i>com</i>	NP-c	P-c	P-c	in L	NP-c
<i>resGr</i>	NP-c	coNP-c	coNP-c	P-c	in P
<i>prf</i>	NP-c	Π_2^P -c	in Θ_2^P	coNP-c	NP-c
<i>sem</i>	Σ_2^P -c	Π_2^P-c	Π_2^P-c	coNP-c	NP-c
<i>stg</i>	Σ_2^P-c	Π_2^P-c	Π_2^P-c	coNP-c	in L

($Exists_\sigma^{-0}$). Des Weiteren betrachten wir noch den “ideal“ Modus des Schließens, eine Verschärfung des skeptisches Schließen bei dem wir uns für Argumente interessieren die sich durch eine ”admissible“ Menge aus skeptisch akzeptablen Argumenten fundieren lassen.

Diese Arbeit gibt eine umfassende computationale Analyse der obigen Problemstellungen für die zehn populärsten Argumentations-Semantiken, diese sind “conflict-free” (*cf*), “naive“ (*naive*), ”grounded“ (*grd*), ”stable“ (*stb*), “admissible” (*adm*), “complete” (*com*), “resolution-based grounded” (*resGr*), “preferred” (*prf*), “semi-stable” (*sem*) und “stage” (*stg*). Wir gliedern diese Analyse in drei Kapitel, (klassische) Komplexitätsanalyse, Effiziente Lösbarkeit, und Übersetzbarkeit von Argumentations-Semantiken.

3 Komplexitätsanalyse

Die Komplexität dieser Problemstellungen für die verschiedenen Semantiken wurde in der Literatur teilweise schon studiert [CMDM05, DBC02, Dun09, DC08, BDG11], dennoch gibt es viele offene Fragen. Hier komplementieren wir existierende Resultate wie folgt:

- Wir geben exakte Komplexitätsklassifizierungen für die ”semi-stable“ und die ”stage“ Semantik, die zeigen, dass Schlussfolgern mit diesen Semantiken in der 2.Ebene der polynomiellen Hierarchie liegt. Die Komplexitätsklassifizierung der “semi-stable“ Semantik war ein offenes Problem, das von [DC08] aufgeworfen wurde.
- Wir geben eine umfassende Klassifizierung für das Schlussfolgern im ”ideal“ Modus. Einerseits durch generische Resultate und andererseits durch eine konkrete Analyse des “ideal“ Modus für die hier betrachteten Semantiken.
- Klassifizierung der in Polynomialzeit lösbaren Problemstellungen in solche, die Polynomialzeit-vollständig (P-c) sind und solche, die mit logarithmischen Speicheraufwand (in L) berechnet werden können.

Die Resultate sind zusammen mit den Komplexitätsresultaten aus der Literatur in Tabelle 1 zusammengefasst.

4 Effiziente Lösbarkeit

Unsere Komplexitätsanalyse zeigt, dass die meisten Problemstellungen NP-schwer sind und daher ein schlecht skalierendes worst-case Laufzeitverhalten haben. Ein Zugang zu “tractable fragments“ ist es, AFs mit spezieller Graphstruktur zu betrachten. Hierzu gibt es erste Arbeiten für AFs bei denen alle Attacks symmetrisch sind [CMDM05], für AFs ohne Zyklen gerader Länge [DBC02] und eine Analyse verschiedener Ansätze für die “preferred“ Semantik [Dun07]. Alle diese Arbeiten beschränken sich aber auf einige wenige Semantiken. Ein Ansatz für *fixed-parameter tractability* ist es, AFs als gerichtete Graphen aufzufassen und die Vielzahl an Graph-Parametern zu nutzen. Erste Untersuchungen für parametrisierte Komplexität von Argumentationsproblemen gibt es in [Dun07], wo die Baumweite als Parameter und in [OS11], wo die Distanz zu speziellen Graphklassen als Parameter genutzt wird. Wir komplementieren diese Resultate wie folgt:

- Wir studieren vier Graphklassen, nämlich azyklische AFs, AFs ohne gerade Zyklen, bipartite AFs und symmetrische AFs, und untersuchen ob sie “tractable fragments“ für die betrachteten Semantiken sind. Dazu geben wir für die verschiedenen Semantiken und Graphklassen exakte Komplexitätsklassifikationen, zeigen also insbesondere das gewisse Graphklassen keine “tractable fragments“ für gewisse Semantiken sein können. Dabei lösen wir ein offenes Problem aus [BDG11] bezüglich der Komplexität des gutgläubigen Schließens mit der “resolution-based grounded“ Semantik in bipartiten AFs. Des weiteren sei erwähnt, dass die breite Komplexitätsanalyse der obigen Graphklassen das Fundament für die erwähnten Distanz-Parametrierungen stärkt und auch zu einer gemeinsamen Publikation geführt [DOS12] hat.
- Für die Problemstellungen, deren Komplexität auf dem zweiten Level der polynomiellen Hierarchie liegt (also schwerer als NP sind), betrachten wir neben den oben genannten “tractable fragments“ auch Kriterien die diese Probleme NP-einfach machen. Diese Kriterien reduzieren die Komplexität, aber nicht ausreichend um die Probleme in Polynomialzeit zu lösen. Solche Kriterien erlauben es dann aber SAT-Solver für Probleme die im Allgemeinen zu hohe Komplexität haben zu verwenden. Diese Analyse bildet die Grundlage für das CEGARTIX¹ System [DJWW12], das auf iterativen Aufrufen von SAT-Solvern basiert und gegenwärtig das schnellste System für “preferred“, “semi-stable“, und “stage“ Semantiken ist.
- Im Bereich “Fixed-Parameter Tractability“ komplementieren wir einerseits bestehende Resultate, indem wir zeigen, dass alle betrachteten Semantiken “fixed-parameter tractable“ bezüglich Baumweite sind. Andererseits erweitern wir diese Resultate auf den Parameter Cliquesweite, eine Generalisierung der Baumweite.
- Im Bereich “Fixed-Parameter Intractability“ studieren wir Parameter die speziell für gerichtete Graphen entworfen wurden. Wir zeigen, dass alle unsere Problemstellung NP/coNP-schwer bleiben, selbst wenn wir einen der Parameter cycle-rank, gerichtete Pfadweite, Kelly-Weite, DAG-Weite oder gerichtete Baumweite beschränken.

Die Resultate dieses Kapitels sind in Tabelle 2 zusammengefasst.

¹<http://www.dbai.tuwien.ac.at/proj/argumentation/cegartix/>

Tabelle 2: "Tractability" für Abstrakte Argumentation. ✓: die betrachtete Problemstellungen auf der gegebenen Graphklassen sind "tractable"; ✗: mind. eine Problemstellung bleibt NP/coNP-schwer.

	<i>stb</i>	<i>adm</i>	<i>com</i>	<i>resGr</i>	<i>prf</i>	<i>sem</i>	<i>stg</i>
Azyklisch	✓	✓	✓	✓	✓	✓	✓
Ohne gerade Zyklen	✓	✓	✓	✓	✓	✓	✗
Bipartit	✓	✓	✓	✗	✓	✓	✓
Symmetrisch	✗	✓	✓	✓	✓	✗	✗
Beschränkte Baumweite	✓	✓	✓	✓	✓	✓	✓
Beschränkte Cliquesweite	✓	✓	✓	✓	✓	✓	✓
Beschränkter cycle-rank	✗	✗	✗	✗	✗	✗	✗
Beschr. gerichtete Pfadweite	✗	✗	✗	✗	✗	✗	✗
Beschränkte Kelly-Weite	✗	✗	✗	✗	✗	✗	✗
Beschränkte DAG-Weite	✗	✗	✗	✗	✗	✗	✗
Beschr. gerichtete Baumweite	✗	✗	✗	✗	✗	✗	✗

5 Übersetzbarkeit von Argumentations-Semantiken

Eine der populärsten Arten abstrakte Argumentation zu implementieren ist die Übersetzung in einen anderen Formalismus. Anstatt von Grund auf neue Algorithmen zu designen, entwirft man eine Codierung des Argumentationsproblems in einem bekannten Formalismus, für den es bereits ausgereifte Systeme gibt. Beispiele für solche Codierungen sind Übersetzungen in Aussagenlogik [BD04] und Answerset-Programming (ASP) [EGW10]. In dieser Arbeit beschäftigen wir uns mit der Möglichkeit diesen Ansatz innerhalb der abstrakten Argumentation zu verwenden. Wir interessieren also dafür eine Semantik in andere zu übersetzen, indem wir Modifikationen am Argumentation Framework vornehmen. Eine Semantik A kann in eine Semantik B übersetzt, wenn es eine Funktion gibt die ein beliebiges AF so modifiziert, dass die A -kohärenten Mengen des ursprünglichen AFs mit den B -kohärenten Mengen des modifizierten AFs übereinstimmen. Sind die kohärenten Mengen exakt die gleichen sprechen wir von einer exakten Übersetzung, unterscheiden sie sich nur auf neuen Argumenten die durch die Modifikation eingeführt worden sind, stimmen aber auf den ursprünglichen Argumenten überein, von einer sinngetreuen Übersetzung. Wenn die Übersetzung effizient berechnet werden kann genügen beide Arten von Übersetzungen unseren Zwecken. Bei exakten Übersetzungen erhält man die kohärenten Mengen des ursprünglichen Problems unmittelbar, bei sinngetreuen Übersetzungen kann man diese durch eine einfache Projektion erhalten. Das Ziel ist also, wann immer möglich, exakte Übersetzungen zu nutzen (die ja auch sinngetreue Übersetzungen sind) und wenn das nicht möglich ist auf sinngetreue Übersetzungen auszuweichen. Wir betrachten sieben Semantiken² und präsentieren die folgenden Resultate:

- Wir betrachten alle Paare von Semantiken A, B und präsentieren (wenn möglich) effiziente exakte oder sinngetreue Übersetzungen von A nach B .
- Wir zeigen, dass es zwischen bestimmten Semantiken keine effiziente exakte / sinnge-

²Die Semantiken "conflict-free" und "naive" sind an sich schon effizient berechenbar. Die "Resolution-based grounded" Semantik ist auf Grund ihrer andersartigen Natur nicht (direkt) für diesen Ansatz zugänglich.

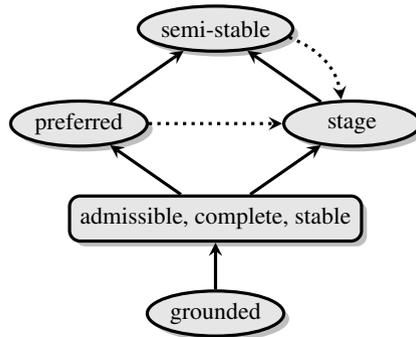


Abbildung 2: Übersetzbarkeit von Argumentations-Semantiken.

treue Übersetzungen geben kann. Manche dieser Resultate basieren auf den präsentierten Komplexitätsresultaten und damit auch auf typischen komplexitätstheoretischen Annahmen, z.B. $P \neq NP$, andere wiederum folgen unmittelbar aus den unterschiedlichen Eigenschaften der Semantiken.

Die konkreten Resultate bezüglich (sinngetreuen) Übersetzungen sind in Abbildung 2 zusammengefasst. Ein durchgezogener Pfeil von Semantik A zu einer Semantik B drückt aus, dass es eine effiziente sinngetreue Übersetzung von A nach B gibt. Aufgrund der Transitivität von Übersetzungen gilt selbiges für Pfade von solchen Pfeilen. Die zwei gepunkteten Pfeile repräsentieren offene Fragen, d.h. die Fälle für die wir weder eine effiziente sinngetreue Übersetzung angeben noch einen Beweis für die Nichtexistenz finden konnten. Wenn es keinen Pfad von A nach B gibt, haben wir bewiesen (möglicherweise unter komplexitätstheoretischen Annahmen), dass es keine effiziente Übersetzung gibt.

6 Conclusio

In dieser Arbeit geben wir eine umfassende Komplexitätsanalyse von Abstrakter Argumentation, sowohl im Sinne der klassischen als auch der parametrisierten Komplexitätstheorie. Diese Resultate zeigen einerseits Möglichkeiten und andererseits Schranken für effiziente Systeme auf. Um der Vielzahl an unterschiedlichen Semantiken Herr zu werden studieren wir das Konzept von Übersetzungen zwischen verschiedenen Semantiken.

Die Relevanz dieser Untersuchungen für das Forschungsgebiet wird durch mehrere wissenschaftliche Publikationen untermauert: Resultate der klassischen Komplexitätsanalyse wurden in [DW10, DDW11] publiziert; "fixed-parameter tractability" Resultate wurden in [DPW12, DSW10] veröffentlicht; und die Übersetzbarkeit von Argumentations-Semantiken wurde in [DW11] publiziert.

Die praktische Bedeutung dieser theoretischen Untersuchungen wird durch zwei auf dieser Arbeit basierenden Systeme belegt. Algorithmen basierend auf dem Parameter Baumweite

werden in [DPW12] präsentiert und sind im dynpartix³ System implementiert. Das schon erwähnte CEGARTIX System nutzt Parametrisierungen der NP-Fragmente von Semantiken höherer Komplexität um diese mit Hilfe von SAT-Solvern zu implementieren.

Weiterführende Forschungsthemen Aufbauend auf den präsentierten Komplexitätsresultaten kann man verallgemeinerte Problemstellungen betrachten. Ein Beispiel hierzu ist das Berechnen des 8-wertigen “justification status” eines Arguments [Dvo12b]. Ein zentraler Punkt um das erwähnte CEGARTIX System weiter zu verbesserten ist das Identifizieren von weiteren NP-Fragmenten und darauf aufbauenden Parametrisierungen. Das Konzept von Übersetzungen zwischen Semantiken ist nicht nur aus computationaler Sicht interessant, sondern erlaubt es auch die Ausdrucksstärke von Semantiken zu vergleichen. Aus dieser Sicht sind dann auch Übersetzungen interessant, die nicht effizient sind. Erste Resultate hierzu wurden in [DS12] veröffentlicht.

Literatur

- [BCD07] Trevor J. M. Bench-Capon and Paul E. Dunne. Argumentation in Artificial Intelligence. *Artif. Intell.*, 171(10-15):619–641, 2007.
- [BCG11] Pietro Baroni, Martin Caminada, and Massimiliano Giacomin. An Introduction to Argumentation Semantics. *Knowledge Eng. Review*, 26(4):365–410, 2011.
- [BD04] Philippe Besnard and Sylvie Doutre. Checking the Acceptability of a Set of Arguments. In *Proc. NMR 2004*, 59–64, 2004.
- [BDG11] Pietro Baroni, Paul E. Dunne, and Massimiliano Giacomin. On the Resolution-Based Family of Abstract Argumentation Semantics and its Grounded Instance. *Artif. Intell.*, 175(3-4):791–813, 2011.
- [CA07] Martin Caminada and Leila Amgoud. On the Evaluation of Argumentation Formalisms. *Artif. Intell.*, 171(5-6):286–310, 2007.
- [CMDM05] Sylvie Coste-Marquis, Caroline Devred, and Pierre Marquis. Symmetric Argumentation Frameworks. In *Proc. ECSQARU 2005*, LNCS 3571, pages 317–328. Springer, 2005.
- [DBC02] Paul E. Dunne and Trevor J. M. Bench-Capon. Coherence in Finite Argument Systems. *Artif. Intell.*, 141(1/2):187–203, 2002.
- [DC08] Paul E. Dunne and Martin Caminada. Computational Complexity of Semi-Stable Semantics in Abstract Argumentation Frameworks. In *Proc. JELIA 2008*, LNCS 5293, pages 153–165. Springer, 2008.
- [DDW11] Wolfgang Dvořák, Paul E. Dunne, and Stefan Woltran. Parametric Properties of Ideal Semantics. In *Proc. IJCAI 2011*, pages 851–856. IJCAI/AAAI, 2011.
- [DJWW12] Wolfgang Dvořák, Matti Järvisalo, Johannes Peter Wallner, and Stefan Woltran. Complexity-Sensitive Decision Procedures for Abstract Argumentation. In *Proc. KR 2012*, pages 54–64. AAAI Press, 2012.

³<http://www.dbai.tuwien.ac.at/research/project/argumentation/dynpartix/>

- [DOS12] Wolfgang Dvořák, Sebastian Ordyniak, and Stefan Szeider. Augmenting Tractable Fragments of Abstract Argumentation. *Artif. Intell.*, 186(0):157–173, 2012.
- [DPW12] Wolfgang Dvořák, Reinhard Pichler, and Stefan Woltran. Towards Fixed-Parameter Tractable Algorithms for Abstract Argumentation. *Artif. Intell.*, 186(0):1 – 37, 2012.
- [DS12] Wolfgang Dvořák and Christof Spanring. Comparing the Expressiveness of Argumentation Semantics. In *Proc. COMMA 2012*, FAIA 245, pages 261–272. IOS Press, 2012.
- [DSW10] Wolfgang Dvořák, Stefan Szeider, and Stefan Woltran. Reasoning in Argumentation Frameworks of Bounded Clique-Width. In *Proc. COMMA 2010*, FAIA, pages 219–230. IOS Press, 2010.
- [Dun95] 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.
- [Dun07] Paul E. Dunne. Computational Properties of Argument Systems Satisfying Graph-Theoretic Constraints. *Artif. Intell.* 171(10-15):701–729, 2007.
- [Dun09] Paul E. Dunne. The Computational Complexity of Ideal Semantics. *Artif. Intell.*, 173(18):1559–1591, 2009.
- [Dvo12] Wolfgang Dvořák. Computational Aspects of Abstract Argumentation. Dissertation, Technische Universität Wien, Institut für Informationssysteme, 2012.
- [Dvo12b] Wolfgang Dvořák. On the Complexity of Computing the Justification Status of an Argument. In *Proc. TFAFA 2011*, LNCS 7132, pages 32–49. Springer, 2012.
- [DW10] Wolfgang Dvořák and Stefan Woltran. Complexity of Semi-Stable and Stage Semantics in Argumentation frameworks. *Inf. Process. Lett.*, 110(11):425–430, 2010.
- [DW11] Wolfgang Dvořák and Stefan Woltran. On the Intertranslatability of Argumentation Semantics. *J. Artif. Intell. Res. (JAIR)*, 41:445–475, 2011.
- [EGW10] Uwe Egly, Sarah Gaggl, and Stefan Woltran. Answer-Set Programming Encodings for Argumentation Frameworks. *Argument and Computation*, 1(2):147–177, 2010.
- [GMAB04] Guido Governatori, Michael J. Maher, Grigoris Antoniou, and David Billington. Argumentation Semantics for Defeasible Logic. *J. Log. Comput.*, 14(5):675–702, 2004.
- [Nie06] Rolf Niedermeier. *Invitation to Fixed-Parameter Algorithms*, volume 31. Oxford University Press, USA, 2006.
- [OS11] Sebastian Ordyniak and Stefan Szeider. Augmenting Tractable Fragments of Abstract Argumentation. In *Proc. IJCAI 2011*, pages 1033–1038. IJCAI/AAAI, 2011.



Wolfgang Dvořák wurde am 24. Juni 1984 in Wien geboren. Er maturierte 2002 am Bundesrealgymnasium Franklinstr. 26 in Wien. Nach dem Grundwehrdienst studierte er an der Technischen Universität Wien das Diplomstudium Technische Mathematik und das Masterstudium Computational Intelligence. Von 2006 – 2012 war er Mitglied der Forschungsgruppe für Datenbanken und Artificial Intelligence an der TU Wien an der er von 2009 bis 2012 in Informatik promovierte. Seit 2012 ist er an der Fakultät für Informatik der Universität Wien in der Forschungsgruppe “Theorie und Anwendungen von Algorithmen” tätig.