Die Komplexität der Formelauswertung in intuitionistischen Logiken

Dissertation
zur Erlangung des akademischen Grades
doctor rerum naturalium (Dr. rer. nat.)

vorgelegt dem Rat der
Fakultät für Mathematik und Informatik
der Friedrich-Schiller-Universität Jena

von Diplom-Mathematiker Felix Weiß
geboren am 12. Januar 1983 in Jena

Gutachter:

1. Prof. Dr. Martin Mundhenk, Friedrich-Schiller-Universität Jena
2. Prof. Dr. Heribert Vollmer, Gottfried Wilhelm Leibniz Universität Hannover

Tag der öffentlichen Verteidigung: 10.01.2013

Zusammenfassung

Der Intuitionismus ist eine Denkweise, die auf Intuition und Konstruktivismus basiert. Dabei sollen Sachverhalte direkt erkennbar sein. Mathematisch gesehen werden im Intuitionismus nur Beweise anerkannt, die konstruktiv sind. Dies schließt zum Beispiel Widerspruchsbeweise aus. Die intuitionistische Logik greift diesen Gedanken in der Form auf, dass das Gesetz des ausgeschlossenen Dritten hier keine Gültigkeit besitzt. Eine Aussage „*A oder nicht A*“ gilt nur dann als wahr, wenn entweder A oder das Gegenteil von A bewiesen werden kann. Der Begriff der Wahrheit wird durch „*beweisbar*“ ersetzt.
Für die intuitionistische Aussagenlogik gibt es eine Semantik, die jener der Modallogiken sehr ähnlich ist. In diesem Sinne kann man die intuitionistische Aussagenlogik auch als spezielle Modallogik auffassen.
Wir beschäftigen uns in dieser Arbeit im Wesentlichen mit der Formelauswertung in intuitionistischen Logiken und untersuchen ihre Komplexität. Dabei betrachten wir verschiedene Fragmente, die durch unterschiedliche Einschränkungen entstehen. Solche Einschränkungen gibt es zum einen auf der semantischen Seite in Form von Beschränkungen der zugelassenen Modelle und zum anderen auf der syntaktischen Seite. Hier kann man die Zahl der Variablen beschränken oder nur bestimmte Operatoren zulassen.
Unsere ersten Ergebnisse beziehen sich auf Logiken, welche derart eingeschränkt wurden, dass es nur noch endlich viele paarweise nicht äquivalente Formeln gibt. Hier können wir zeigen, dass neben der Formelauswertung auch das Erfüllbarkeits- und das Tautologieproblem sehr einfach zu lösen sind. Danach betrachten wir die Logik, bei der nur eine Variable zugelassen ist. Für diese Logik zeigen wir, dass die Formelauswertung AC^1-vollständig ist. Dies ermöglicht eine neue Sicht auf die Klasse AC^1, da sie bisher nur durch Probleme charakterisiert wurde, welche die speziellen Eigenschaften dieser Klasse selbst haben. Wir grenzen dieses Resultat ab, indem wir zeigen, dass es maßgeblich von der Zahl der Variablen, der Art der Formeldarstellung und den zugelassenen Modellen abhängt. Weiter betrachten wir Logiken, deren Formelauswertungsproblem die maximale Komplexität erreicht. Auch hier geht es wieder um Abgrenzung – also um die Frage, welche Freiheitsgrade man in einer Logik mindestens braucht, damit die Formelauswertung derart komplex ist. Am Ende dieser Arbeit betrachten wir noch einige Modallogiken, die Begleiter von intuitionistischen Logiken sind. Uns interessiert hier vor allem das Verhältnis zwischen der Formelauswertung in intuitionistischen Logiken und in ihren modalen Begleitern.
Schlagworte: Intuitionistische Logik, Formelauswertung, Komplexität, Modallogik, AC^1

Danksagungen

An erster Stelle möchte ich meinem Doktorvater Martin Mundhenk für die phantastische Unterstützung in den letzten vier Jahren danken. Ohne ihn wäre diese Arbeit nicht möglich gewesen. Er weckte bereits während meines Studiums mein Interesse an der Komplexitätstheorie und an außergewöhnlichen Logiken und zeigte mir, wie man wissenschaftlich arbeitet. Seine Tür stand mir jederzeit für gemeinsame fachliche Diskussionen und den Austausch von Ideen offen. Oftmals waren es gerade die Anregungen und Denkanstöße von ihm, welche mir bei der Entwicklung einer Lösung maßgeblich halfen.
Besonderer Dank geht auch an Thomas Schneider und Arne Meier, die mir bei Fragen zum wissenschaftlichen Arbeiten und zum Umgang mit LaTeX stets tatkräftig zur Seite standen.
Weiter möchte ich mich bei Robert Zeranski bedanken, mit dem ich zwei Jahre ein Büro teilte. In vielen Gesprächen sind neue Ideen zustande gekommen, wir entwickelten gemeinsam Lösungsansätze und diskutierten kritische Rückfragen. Klaus Reinhardt möchte für die Gespräche über die Logiken PrL, S4.3 und LC danken. Michael Elberfeld trug mit seinen Erklärungen zur Klasse AC^0 wesentlich zu den Resultaten für die endlich erzeugten Logiken bei, dafür vielen Dank. Außerdem geht ein besonderer Dank an Robert Faßler, der diese Arbeit akribisch gelesen und kontrolliert hat.
Natürlich möchte ich auch meinen Eltern Viola und Martin Weiß herzlich dafür danken, dass sie mir mein Studium der Mathematik ermöglicht haben und mir auch in schwierigen Phasen immer mit Rat und Tat zur Seite standen. Ohne diese Unterstützung wäre eine Promotion unmöglich gewesen. Außerdem danke ich meiner Mutter für die zahlreichen Hinweise und die viele Zeit, die sie mit dem Lesen der Arbeit verbracht hat.
Ein weiterer Dank geht an meine Freundin Sybille Meißner, die mich gerade im letzten Jahr, in dem ich aus den Ergebnissen diese Arbeit zusammengeschrieben habe, immer unterstützt hat und mir sehr viel Verständnis entgegen brachte.

Inhaltsverzeichnis

Abbildungsverzeichnis

Kapitel 1

Einleitung

Die Komplexitätstheorie ist ein wesentlicher Bestandteil der theoretischen Informatik. Ihr Hauptziel besteht in der Analyse des Aufwands, den man zur Lösung eines Problems benötigt. Aber was bedeutet es überhaupt, ein Problem zu lösen und wie misst man den Aufwand? Grundsätzlich sollen Probleme algorithmisch gelöst werden. Das heißt, es gibt eine Eingabe, die von einem Algorithmus verarbeitet wird und nach Abschluss der Berechnungen gibt der Algorithmus eine Ausgabe (die Lösung) aus. Dabei unterscheidet man zwei Typen von Problemen, das *Entscheidungsproblem*, bei dem die Ausgabe immer nur „Ja“ oder „Nein“ ist, und das *Berechnungsproblem*, bei dem ein Funktionswert als Ausgabe berechnet wird. Wir beschränken uns in dieser Arbeit auf Entscheidungsprobleme. Bei solchen Problemen sagt man auch, dass eine Eingabe entweder *akzeptiert* (Ausgabe „Ja“) oder *abgelehnt* (Ausgabe „Nein“) wird. Der Begriff *Aufwand* fasst verschiedene Aspekte zusammen. Es geht dabei unter anderem um die Ressourcen, welche für die Lösungsbestimmung benötigt werden – wie lange läuft ein Algorithmus und wie viel Speicherplatz braucht er? Außerdem spielt auch die Art des Algorithmus eine Rolle – ist er deterministisch oder nichtdeterministisch, enthält er Alternierungen oder nutzt er ein Orakel? Die *Komplexität* dient als Maß für diesen Aufwand. Das Ziel besteht immer darin, für ein Problem zu untersuchen, ob es für eine Komplexitätsklasse *vollständig* ist. Vollständigkeit bedeutet, dass es sich mit den Ressourcen lösen lässt, die in dieser Klasse zur Verfügung stehen – also in der Klasse *enthalten* ist – und dass diese Ressourcen für die Lösung auch notwendig sind – das Problem ist dann *hart* für diese Klasse. Hat man in der Praxis ein schwer zu lösendes Problem, so kann man sich die Frage stellen, ob man das komplette Problem bearbeiten will, oder ob es möglich ist, gewisse Einschränkungen in Kauf zu nehmen. Oftmals lässt sich eine eingeschränkte Variante des Problems deutlich leichter lösen.

Ein Beispiel ist die Färbung von Graphen. Das Ziel dabei ist, für einen gegebenen Graphen zu entscheiden, ob es möglich ist, die Knoten mit verschiedenen Farben so zu färben, dass benachbarte Knoten niemals dieselbe Farbe haben. Die Anzahl der Farben ist dabei fest vorgegeben. Ein sehr einfacher Ansatz besteht darin, für alle möglichen Färbungen der Knoten zu überprüfen, ob (mindestens) eine im obigen Sinne korrekt ist. Das Problem bei dieser Variante steckt in „. . . alle möglichen. . . “, denn für einen Graphen mit n Knoten, der mit k Farben gefärbt werden

soll, gibt es k^n verschiedene Färbungen. Bei einem Graphen mit 25 Knoten muss man bei drei Farben bereits knapp 850 Milliarden Färbungen testen. Die Laufzeit eines solchen naiven Algorithmus hängt offensichtlich exponentiell von der Größe des Graphen ab. Es gibt sicher deutlich effizientere Algorithmen, aber ein wirklich schneller Ansatz ist bisher nicht bekannt.[1] Beschränkt man sich bei der Anzahl der Farben allerdings auf zwei, so bekommt man einen sehr einfachen und sehr schnellen Algorithmus: Man färbt einen Knoten in der einen Farbe und alle Nachbarn bekommen die andere Farbe, alle noch nicht gefärbten Nachbarn bekommen wieder die erste Farbe und so weiter. Muss man einen Knoten in einer Farbe färben, die ein Nachbar bereits hat, weiß man, dass sich dieser Graph nicht mit zwei Farben färben lässt. Dieser Algorithmus benötigt offensichtlich viel weniger Rechenzeit als die erste Variante. Aber dafür gibt es die Einschränkung, dass man das Problem nur noch für zwei Farben untersuchen kann. Bereits für drei Farben ist das Problem NP-hart – also nicht mehr deterministisch in polynomieller Zeit lösbar (außer im unwahrscheinlichen Fall, dass $\mathsf{P} = \mathsf{NP}$ gilt).

Auch die Logik spielt in der Komplexitätstheorie eine große Rolle. Das wohl bekannteste Entscheidungsproblem der Logik ist das Erfüllbarkeitsproblem der Aussagenlogik SAT. Dabei ist eine aussagenlogische Formel gegeben und die Frage lautet, ob es für diese Formel eine Belegung der Variablen gibt, die sie *erfüllt* – also, ob der Wahrheitswert der gesamten Formel unter dieser Belegung *wahr* ist. Für dieses Problem hat Cook bereits 1971 die NP-Härte gezeigt [Coo71b]. In sehr vielen NP-Härte-Beweisen spielt SAT eine zentrale Rolle. Die NP-Härte des 3-Färbbarkeitsproblems lässt sich ebenfalls leicht mit Hilfe von SAT zeigen. Auch von SAT gibt es verschiedene Varianten, unter anderem k-SAT, bei dem man keine beliebigen Formeln untersucht. Hier werden nur Formeln in konjunktiver Normalform betrachtet, wobei die einzelnen Klauseln aus höchstens k Literalen bestehen. Für jede Formel der Aussagenlogik gibt es eine äquivalente Formel in konjunktiver Normalform. Auch hier ist der Sprung von polynomieller Laufzeit zu NP beim Übergang von $k = 2$ zu $k = 3$. Während 3-SAT NP-hart ist, lässt sich 2-SAT in polynomieller Laufzeit lösen.

In erster Linie geht es in dieser Arbeit um die Betrachtung des Formelauswertungsproblems für verschiedene intuitionistische Logiken unter komplexitätstheoretischen Gesichtspunkten. Ein wichtiges Werkzeug dafür sind alternierende Pfade in Schichtgraphen, da zwischen ihnen und alternierenden Berechnungen im Allgemeinen ein enger Zusammenhang besteht. Wir nutzen diesen Zusammenhang für einen großen Teil unserer Härte-Resultate. Die oberen Schranken – also die Antwort auf die Frage, in welcher Komplexitätsklasse ein Problem enthalten ist – geben wir meist mit Hilfe von Algorithmen an. Neben der Einordnung in Komplexitätsklassen interessiert uns auch der Sprung zwischen polynomieller Laufzeit und effizienter Parallelisierbarkeit. Ein Problem ist effizient parallelisierbar, wenn

[1] Das 3-Färbbarkeitsproblem spielt in dieser Arbeit keine weitere Rolle, Details dazu und eine Einführung in die Graphentheorie können [Die06] entnommen werden. Approximationsansätze gibt es unter anderem von Blum [Blu92].

man einen Algorithmus angeben kann, der logarithmische Laufzeit benötigt, aber Berechnungen parallel (bzw. alternierend) durchführen kann. Insbesondere interessiert uns hier, welche Einschränkungen wir bei unseren Problemen vornehmen müssen, um diesen Sprung zu schaffen. Dabei wollen wir auch zeigen, dass unsere Ergebnisse optimal sind, das heißt, wir zeigen, dass die Einschränkungen nicht weiter abgeschwächt werden können.

1.1 Logik

Die Basis der Logiken in dieser Arbeit bildet die Aussagenlogik. In der Aussagenlogik lassen sich nur sehr einfache Zusammenhänge formulieren. Die grundlegenden Bestandteile sind atomare Aussagen wie „der Apfel ist rot“ oder „der Mond ist ein Käsekuchen“. Dazu kommen logische Operatoren wie *und* ($\wedge$), *oder* ($\vee$) und *nicht* ($\neg$). Um über komplexere Zusammenhänge sprechen zu können, muss man die Logik erweitern. Eine solche sehr umfangreiche Erweiterung ist die Prädikatenlogik. Auf der einen Seite kann man mit der Prädikatenlogik sehr viel mehr ausdrücken als mit der Aussagenlogik, auf der anderen Seite führt diese gewonnene Ausdrucksstärke aber dazu, dass Berechnungen sehr viel komplizierter und teilweise sogar unmöglich werden.
Während sich das Formelauswertungsproblem in der Aussagenlogik sehr leicht lösen lässt (in NC^1 [Bus87]), ist es in der Prädikatenlogik mit herkömmlicher Rechentechnik im Prinzip gar nicht mehr lösbar (PSPACE-hart [Sto74]). Das Erfüllbarkeitsproblem der Aussagenlogik lässt sich nichtdeterministisch in polynomieller Zeit lösen [Coo71b], in der Prädikatenlogik ist es nicht entscheidbar [Chu36, Tur37]. Es kann also durchaus sinnvoll sein, sich Logiken mit einer geringeren Ausdrucksstärke anzuschauen. Eine solche Logik, die zwischen Aussagen- und Prädikatenlogik liegt, ist zum Beispiel die Modallogik. Formeln lassen sich in der Modallogik in polynomieller Zeit auswerten [FL79] und ihr Erfüllbarkeitsproblem ist immerhin noch in polynomiellem Platz berechenbar [Lad77].

1.1.1 Modallogik

In der Modallogik spielen neben *wahr* und *falsch* auch die Begriffe *möglich* und *notwendig* eine Rolle. Sie wurde 1918 durch Lewis [Lew18] eingeführt und ist eine Erweiterung der Aussagenlogik. Zusätzlich zur Aussagenlogik werden in der Modallogik noch die beiden Operatoren *Diamant* ($\Diamond$) und *Box* ($\Box$) verwendet.[2] Die Box drückt die Notwendigkeit aus, der Diamant die Möglichkeit. Um dieser Logik Anschaulichkeit zu verleihen, entwickelte Kripke 1963 eine relationale Semantik [Kri63a]. Die Kripke-Semantik basiert auf Graphen, deren Knoten *Welten* sind und deren Kanten die Verbindungen zwischen den Welten darstellen.

[2]Tatsächlich benötigt man nur einen Operator, denn der andere kann simuliert werden. Aus Gründen der besseren Anschaulichkeit werden aber oft beide verwendet.

Durch die moderne Rechentechnik kommt dieser Semantik eine besondere Bedeutung zu: Den Ablauf eines Programms auf einer Maschine kann man als eine Reihe von Zuständen und Zustandsübergängen auffassen. Diese Zustände und Übergänge lassen sich in einer relationalen Struktur sehr gut darstellen. Darüber hinaus eignet sich die Modallogik zur Darstellung und Modellierung zeitlicher Abläufe im Allgemeinen oder der Verteilung (und Entwicklung) von Wissen in Personengruppen. Auf Basis der Idee der allgemeinen Modallogik entstanden viele weitere Modallogiken, die auch als *Fragmente* bezeichnet werden. Dabei gibt es grundsätzlich zwei verschiedene Arten, diese Fragmente zu bilden. Bei der einen Variante sind die zu verwendenden Operatoren eingeschränkt, die Grundlage hierfür liefert der Postsche Verband [Pos41]. Die Fragmente der anderen Variante entstehen durch Hinzunahme von neuen Axiomen oder – betrachtet man die Logik aus Sicht der Kripke-Semantik – durch eine Einschränkung der Modelle. Untersuchungen solcher Fragmente sind unter anderem in [Lad77, HM92, Spa93, Sch02] zu finden. In dieser Arbeit betrachten wir vor allem Fragmente, welche durch eine Kombination beider Varianten entstehen. Auch die intuitionistische Aussagenlogik kann als ein solches Fragment aufgefasst werden.
Seit der Einführung der Kripke-Modelle 1963 [Kri63a] wurde die Modallogik unter vielen Aspekten untersucht. Das Buch *Handbook of Modal Logic* von Blackburn et al. [BvBW06] bietet eine gute und sehr umfangreiche Übersicht. Einen Überblick über die Entwicklung der Erforschung gibt Goldblatt 2006 [Gol06]. Auch komplexitätstheoretische Gesichtspunkte spielen bei den Untersuchungen immer wieder eine große Rolle. Eines der wichtigsten Resultate stammt von Ladner 1977 [Lad77]. Er zeigte, dass Tautologie- und Erfüllbarkeitsproblem beide PSPACE-vollständig sind. Für die Fragmente S4 und S4.2 konnte dieses Resultat ebenfalls gezeigt werden (siehe [Spa93]). Interessanterweise gilt das Ergebnis auch dann noch, wenn man sich auf die Fragmente dieser Logiken beschränkt, die nur mit strikter Implikation als Operator auskommen [Bou04]. Fragmente mit beschränkter Variablenzahl werden in [Sve03a, CR02] untersucht. Bisher sind keine Resultate zur PSPACE-Härte für modale Fragmente mit strikter Implikation und einer beschränkten Variablenzahl bekannt. Das Erfüllbarkeitsproblem untersuchten Hemaspaandra et al. [HSS10] für alle Fragmente bezüglich des Postschen Verbandes vollständig. Eine allgemeine Arbeit zu diesem Thema gibt es von Halpern und Moses 1992 [HM92]. Sie beschäftigen sich unter anderem mit grundlegenden Techniken zur Bestimmung von unteren und oberen Komplexitätsschranken.

1.1.2 Intuitionistische Logik

Gibt es zwei irrationale Zahlen a und b, so dass a^b rational ist? Angenommen $\sqrt{2}^{\sqrt{2}}$ ist rational. Dann können wir die Frage mit „Ja“ beantworten, denn $\sqrt{2}$ ist bekanntlich irrational. Ist hingegen $\sqrt{2}^{\sqrt{2}}$ nicht rational, können wir die Frage auch mit „Ja“ beantworten, denn für $a = \sqrt{2}^{\sqrt{2}}$ und $b = \sqrt{2}$ ist a^b rational. Also haben wir bewiesen, dass es zwei solche Zahlen gibt, jedoch ohne ganz konkrete Vertreter angeben zu können. Diese Art von Beweis ist nicht konstruktiv und genau solche

Beweise werden im Intuitionismus abgelehnt.
Der Intuitionismus, wie wir ihn verwenden, wurde von Brouwer in den 20er und 30er Jahren des letzten Jahrhunderts begründet. Der Grundgedanke ist grob gesprochen, dass eine Wahrheit nur als Wahrheit zählt, wenn sie konstruktiv bestimmt wurde. In diesem Sinne werden die Begriffe *Wahrheit* und *beweisbar* gleich gesetzt – es sind nur Aussagen wahr, die durch einen konstruktiven Beweis gezeigt werden können. Nimmt man Konstruierbarkeit als Voraussetzung für Existenz, so verliert man unmittelbar den Satz des ausgeschlossenen Dritten. Gibt es weder für eine Aussage noch für deren Gegenteil einen Beweis, so gelten beide im intuitionistischen Sinne als nicht wahr. Ein konstruktiver Beweis für eine Disjunktion enthält immer einen Beweis für einen der beiden Teile.
Anfänglich gestaltete Brouwer die Mengenlehre nach intuitionistischen Prinzipien [Bro18]. Die Grundlagen der intuitionistischen Logik gehen aus einer Arbeit von 1925 [Bro25] hervor. Er lehnte das Prinzip des ausgeschlossenen Dritten ab und zeigte, dass dann die doppelte Negation einer Aussage nicht die Aussage selbst ist. Er behielt bei, dass aus einer Aussage ihre doppelte Negation folgt. Daraus ergab sich die Äquivalenz zwischen dreifacher und einfacher Negation. Sein Schüler Heyting befasste sich ebenfalls mit der intuitionistischen Logik und gab 1930 das erste formalisierte Axiomensystem an [Hey30, Hey86]. Eine einfache Formalisierung der intuitionistischen Aussagenlogik kann man auf Basis von Gentzens Kalkül [Gen34] des natürlichen Schließens angeben. Gegenüber dem natürlichen Schließen der Aussagenlogik besteht hier der Unterschied darin, dass die Regel *Reductio ad absurdum* nicht gilt. Anschaulich gesprochen besagt diese Regel, dass man eine Aussage beweisen kann, indem man ihr Gegenteil widerlegt. Dieses Prinzip entspricht aber gerade nicht dem konstruktiven Ansatz des Intuitionismus. Diese einfache Modifikation des Kalküls macht die Nähe zur normalen Aussagenlogik deutlich.
Den ersten Ansatz für eine Interpretation lieferten Heyting 1931 [Hey31, Hey34] und Kolmogorov 1932 [Kol32]. Diese Interpretation wurde auch Brouwer-Heyting-Kolmogorov-Interpretation oder BHK-Interpretation genannt. Während klassische Interpretationen beschreiben, wie man Wahrheitswerte berechnet, geht es bei der BHK-Interpretation mehr um eine Beschreibung, wie man logisch zusammengesetzte Aussagen beweist, indem man die einzelnen Teile beweist – also wieder um den konstruktiven Gedanken. Die Beschreibung dieser Interpretation blieb so vage, dass sie sich nicht durchgesetzt hat. Einen anderen Ansatz wählte 1936 Jaskowski [Jas36]. Er ging in die algebraische Richtung und verwendete verschiedene Wahrheitswerte. Für diese Semantik, die einen Spezialfall der Heyting-Algebren darstellt, konnte er die Vollständigkeit für die intuitionistische Aussagenlogik beweisen. Auch Stone [Sto37] näherte sich 1937 von der algebraischen Seite. 1938 gab Tarski [Tar38] eine vollständige Semantik an, die auf offenen Mengen in topologischen Räumen basiert (siehe auch [MT44]). Auf dieser Idee beruht auch die Semantik von Beth [Bet47, Bet56]. Dieser Ansatz versucht, die Anschauung umzusetzen, dass Wissen durch immer neue Beweise „wächst“. Mit endlich verzweigten Bäumen wird die Inklusionsstruktur von offenen Mengen dargestellt. In

diesem Sinne führte er bereits eine Vorform der Kripke-Semantik ein. Kripke definierte mit den Kripke-Modellen zuerst eine Semantik für die Modallogik [Kri63a] und konnte diese dann durch geringe Modifikationen auch für die intuitionistische (Aussagen-)Logik verwenden [Kri63b, Kri65]. Die Zustände der Kripke-Modelle, die das Wissen repräsentieren, erfüllen jetzt eine Monotonieeigenschaft, die garantiert, dass das Wissen nicht abnimmt. Auch hier ist der intuitionistische Grundsatz anschaulich umgesetzt: Was einmal bewiesen ist, bleibt immer richtig. Neben der Semantik basierend auf den Heyting-Algebren bilden die Kripke-Modelle heute die Standardsemantik für die intuitionistische Logik. Unsere Resultate beziehen sich alle auf die Kripke-Modelle als Semantik, da das Formelauswertungsproblem (wie wir es betrachten) auf den algebraischen Semantiken nicht definiert werden kann. In den algebraischen Semantiken geht es eher um Äquivalenzen zwischen Formeln. Bei den Kripke-Modellen steht das Erfüllen von Formeln im Vordergrund.

Interessant ist, wie sich die intuitionistische Aussagenlogik zwischen der normalen Aussagenlogik und der Modallogik positioniert. Wie bereits erwähnt, ergibt sich die Nähe zur Aussagenlogik in der Syntax, denn es wird dieselbe Formelmenge verwendet.[3] Außerdem muss man bei der syntaktischen Betrachtung auf Basis des natürlichen Schließens nur geringfügige Änderungen vornehmen. Hier gelingt der Übergang von Aussagenlogik zur intuitionistischen Aussagenlogik durch das Entfernen der Regel Reductio ad absurdum. Betrachtet man hingegen die semantische Seite, so nimmt man die Modallogik als Basis, verändert die Kripke-Modelle geringfügig und erhält wieder die intuitionistische Aussagenlogik. In dieser Arbeit betrachten wir alle Logiken von der semantischen Seite. Gödel gab bereits 1932 eine Einbettung der intuitionistischen Aussagenlogik in die Modallogik an [Göd32], allerdings zu diesem Zeitpunkt noch ohne Verwendung der Kripke-Modelle.

Seit den 60er Jahren des letzten Jahrhunderts wurden verschiedene modifizierte Varianten und Fragmente der intuitionistischen Aussagenlogik untersucht. Eine für unser Kapitel 4 wesentliche Arbeit wurde 1960 von Nishimura [Nis60] veröffentlicht. Er untersuchte die intuitionistische Aussagenlogik mit nur einer Variablen auf Basis von Heyting-Algebren und konnte zeigen, dass es im Gegensatz zur normalen Aussagenlogik mit einer Variablen in dieser Logik bereits unendlich viele Formeläquivalenzklassen gibt. Visser konstruierte 1980 [Vis80] die Modifikationen BPL (Basic Propositional Logic) und FPL (Formal Propositional Logic) und gab die Einbettung in spezielle Modallogiken an. Weitere Fragmente sind die superintuitionistischen Logiken KC [DL59] und LC [Dum59].

In der jüngeren Zeit wurden diese intuitionistischen Logiken auch unter komplexitätstheoretischen Gesichtspunkten untersucht. Da sich das Erfüllbarkeitsproblem häufig auf das der normalen Aussagenlogik zurückführen (NP-vollständig) oder sehr einfach lösen lässt (in NC^1), konzentrierte sich die Forschung bisher weitestgehend auf das Tautologieproblem. Das erste Resultat gab 1979 Statman an [Sta79]. Er zeigte, dass das Tautologieproblem für die intuitionistische Aussagen-

[3] Wir unterscheiden in dieser Arbeit aus Gründen der Übersichtlichkeit zwischen intuitionistischer und normaler Implikation.

logik PSPACE-vollständig ist (siehe auch [Cha85, Sve03b]). Eine weitere Arbeit dazu stammt von Rybakov [Ryb06]. Darin untersuchte er, wie viele Variablen für die PSPACE-Härte des Tautologieproblems in den verschieden Logiken ausreichen. Diese Arbeit bildet eine wesentliche Grundlage für die Untersuchungen in Kapitel 5. Es wurde auch für das Fragment, in dem die Implikation der einzige Operator ist, gezeigt, dass das Tautologieproblem PSPACE-vollständig ist [Sta79, Cha85, Sve03b].
Grundsätzlich lässt sich der intuitionistische Gedanke natürlich nicht nur auf die Aussagenlogik übertragen. Da wir uns in dieser Arbeit aber ausschließlich mit der aussagenlogischen Variante beschäftigen, meinen wir immer die intuitionistische Aussagenlogik, wenn wir über intuitionistische Logik sprechen.

1.1.3 Das Formelauswertungsproblem

Das Formelauswertungsproblem bezeichnet allgemein die Frage, ob eine Formel von einem Modell erfüllt wird. Dies kann in der Praxis zum Beispiel genutzt werden, um zu verifizieren, dass ein Programm eine Maschine richtig steuert. Die verschiedenen Zustände der Maschine sind das Modell und das Programm wird durch eine Formel dargestellt. In der Theorie gibt es verschiedene Varianten dieses Problems. Wir betrachten hier den Fall, bei dem sowohl die Formel als auch das Modell Bestandteil der Eingabe sind. Bei einer anderen Variante ist eine bestimmte Formel ein Teil des Problems und die Eingabe besteht nur aus einem Modell, für das gefragt wird, ob es diese Formel erfüllt. Diese Variante kommt hauptsächlich bei der Deskriptiven Komplexität zur Anwendung (siehe [Imm79, Imm89, Imm99]).
In der klassischen Aussagenlogik ist dieses Problem sehr einfach lösbar (alternierend in logarithmischer Zeit [Bus87]), da die Modelle lediglich eine Belegung der Variablen mit Wahrheitswerten sind. Auch für andere Logiken wurde die Komplexität der Formelauswertung bereits untersucht. Als Beispiele sind hier die *Linear Time Logic* LTL [SC85, Mar04, BMS$^+$11] und die *Computation Tree Logic* CTL [MMTV09, Mei11] zu nennen. Diese temporalen Logiken sind spezielle Modallogiken, die Komplexität der Formelauswertung bewegt sich zwischen NC^1 und P (für CTL* bis PSPACE). Für die hybriden Logiken, die eine Erweiterungen der Modallogik sind, erstrecken sich die Komplexitätsresultate von P bis PSPACE [tCF05, FdR06, Sch07a]. Auch für die Prädikatenlogik wurde das Problem bereits untersucht und ist im allgemeinen Fall ebenfalls PSPACE-vollständig [Sto74].
Aus Sicht der Kripke-Modelle kann man Belegungen von atomaren Aussagen mit Wahrheitswerten als Kripke-Modell mit nur einer Welt auffassen. In diesem Sinne ist zu erwarten, dass die Formelauswertung in Logiken, deren Semantik auf allgemeinen Kripke-Modellen basiert, aufwändiger ist als in der Aussagenlogik. Aus [FL79] folgt, dass das Formelauswertungsproblem für alle Logiken in dieser Arbeit in P ist. Wir untersuchen unter anderem, unter welchen Bedingungen das Problem für eine Logik auch P-hart ist und wann es sich mit geringerem Aufwand lösen lässt. Hierbei interessieren uns besonders die Fragmente, für die sich die Formelauswertung effizient parallelisieren lässt (also in NC^1 ist).

Es fällt schwer, praktische Anwendungen für die Formelauswertung in der intuitionistischen Logik zu finden. Aber es geht in dieser Arbeit auch nicht um die praktische Relevanz der Ergebnisse, sondern um die Untersuchung dieses Problems und die möglichst vollständige Charakterisierung für verschiedene Logiken. Dabei ermöglichen die Resultate eine neue Sicht auf die Logiken und ihre Ausdrucksmöglichkeiten. Insbesondere die Logik IPL[1], für deren Formelauswertungsproblem wir AC^1-Vollständigkeit zeigen, erweitertet die Sicht auf eine Komplexitätsklasse, deren vollständige Probleme (nach unserer Kenntnis) bisher immer die speziellen Eigenschaften der Klasse als Teil der Definition enthalten. In diesem Sinne ist das Formelauswertungsproblem für IPL[1] das erste natürliche vollständige Problem für diese Klasse.

1.2 Aufbau der Arbeit und Resultate

In Kapitel 2 definieren wir die zentralen Begriffe dieser Arbeit. Dabei beginnen wir mit der Modallogik und führen die intuitionistische Logik auf Basis der Modallogik ein (Abschnitt 2.1). In Abschnitt 2.2 gehen wir auf die Komplexitätstheorie ein und geben die wesentlichen Komplexitätsklassen und Reduktionsbegriffe an. Zusätzlich stellen wir einige vollständige Probleme vor, die wir später als Werkzeug nutzen. Die Probleme, die in dieser Arbeit untersucht werden, definieren wir in Abschnitt 2.3. Die Resultate dieser Arbeit sind in den folgenden Kapiteln zu finden.
In Kapitel 3 betrachten wir endlich erzeugte Logiken. Für diese Logiken können wir zeigen, dass ihr Formelauswertungsproblem grundsätzlich in NC^1 liegt. Dieses Ergebnis liefert einige Nebenresultate, die wir in Abschnitt 3.2 vorstellen. In Abschnitt 3.3 geben wir einige endlich erzeugten Logiken an. Insbesondere zeigen wir für die Logik LC, dass diese Logik bei Beschränkung der Anzahl der Variablen auf eine feste Zahl endlich erzeugt ist.
In Kapitel 4 geht es um die Logik IPL mit einer Variablen. Wir können hier zeigen, dass diese Logik ein AC^1-vollständiges Formelauswertungsproblem hat (Abschnitte 4.2 und 4.3). Außerdem betrachten wir in Abschnitt 4.4 superintuitionistische Logiken mit einer Variablen und zeigen, dass diese endlich erzeugt sind. Im Abschnitt 4.5 untersuchen wir verschiedene Varianten von IPL mit einer Variablen. Dabei schränken wir die Modelle ein und ändern die Kodierung der Formeln – immer mit der Folge, dass das Formelauswertungsproblem nicht mehr AC^1-vollständig ist.
Intuitionistische Logiken mit P-hartem Formelauswertungsproblem betrachten wir Kapitel 5. In Abschnitt 5.1 beweisen wir für verschiedene Fragmente die P-Härte und in Abschnitt 5.2 zeigen wir, dass diese Ergebnisse in Bezug auf die Zahl der verwendeten Variablen optimal sind.
Das Kapitel 6 beschäftigt sich mit Modallogiken. Dabei konzentrieren wir uns im Wesentlichen auf die Logiken, welche modale Begleiter der von uns untersuchten intuitionistischen Logiken sind (Abschnitt 6.1). Einen Vergleich zwischen intuitionistischen Logiken und ihren modalen Begleitern geben wir in Abschnitt 6.2 an.

Am Ende jedes Kapitels gibt es eine Zusammenfassung, in der wir einen Überblick über die Resultate aus diesem Kapitel geben. In Kapitel 7 fassen wir nochmal alle Resultate kurz zusammen.

1.3 Publikationen

Große Teile dieser Arbeit wurden bereits veröffentlicht. Unsere Resultate zur alternierenden Wegsuche (Theoreme 2.21 und 2.22) in Abschnitt 2.2.2 wurden (teilweise ohne ausführliche Beweise) in [MW10] und [MW11] veröffentlicht. Die Abschnitte 3.1 und 3.2 enthalten neue Resultate, Abschnitt 3.3 verbessert ein Resultat aus [MW10]. Große Teile von Kapitel 4 stammen aus [MW11] und einer Arbeit, die kurz vor der Veröffentlichung steht [MW12a]. Die Kapitel 5 und 6 sind im Wesentlichen in [MW12b] erschienen.

Kapitel 2

Grundlagen

In diesem Kapitel soll der Leser mit den wesentlichen Grundlagen dieser Arbeit vertraut gemacht werden. Dabei setzten wir die Vertrautheit mit mathematischen Grundbegriffen, wie Mengen, Funktionen, Relationen, Graphen und Aussagenlogik voraus. Eine grundlegende Einführung in diese Themen gibt es von Schöning in [Sch00, Sch08], von Schmidt und Strohlein [SS89] sowie von Enderton in [End01]. Ebenso setzen wir den Umgang mit dem Landau Symbol (oder auch $\mathcal{O}$-Notation) voraus, siehe dazu auch [CLR90].

Wenn wir einen Ausdruck A durch B definieren, verwenden wir in dieser Arbeit „:=“ und schreiben „$A := B$“. Durch „$A \iff B$“ drücken wir aus, dass A und B äquivalent sind. Schreiben wir „$A \Rightarrow B$“, meinen wir, dass B aus A folgt. Wir definieren 0 als natürliche Zahl und $\mathbb{N} = \{0, 1, 2, \dots\}$ als die Menge der natürlichen Zahlen.

Für eine aussagenlogische Formel φ bezeichne $\mathrm{TF}(\varphi)$ die Menge aller Teilformeln von φ. Dabei gehen wir von einer natürlichen Ordnung aus. Für $\mathrm{TF}(\varphi) = \{\varphi_1, \varphi_2, \dots, \varphi_n\}$ und $1 \leq i, j \leq n$ ist $i \leq j$, falls φ_i in φ_j als Teilformel vorkommt. Intuitiv gesprochen ist die Menge $\mathrm{TF}(\varphi)$ so organisiert, dass kürzere Teilformeln auch kleinere Indizes haben. Mit $|\varphi|$ bezeichnen wir die Länge der Formel φ, wobei wir alle Operatoren und alle Variablen zählen. Aus φ entsteht die Formel $\varphi[\alpha_1/\beta_1][\alpha_2/\beta_2] \dots [\alpha_m/\beta_m]$, indem in φ alle Vorkommen von α_i durch β_i ersetzt werden ($1 \leq i \leq m$). Die hier erklärten Begriffe lassen sich für Formeln der Modallogik (mit modalen Operatoren) und der intuitionistischen Logik in natürlicher Weise erweitern.

Obwohl es in dieser Arbeit nicht um Prädikatenlogik geht, verwenden wir sie an verschiedenen Stellen als Werkzeug. Eine formale Einführung der Prädikatenlogik ist sehr technisch und umfangreich, deswegen verzichten wir hier darauf. Quantoren und Prädikate benutzen wir in der üblichen Art und Weise. Weil die Prädikatenlogik hier im Wesentlichen im Rahmen der deskriptiven Komplexität gebraucht wird, empfiehlt sich als Einführung das erste Kapitel von [Imm99]. Eine weitere umfangreiche Einführung ist in [vD04] zu finden.

Da in Formeln immer geklammert wird und diese Klammerung korrekt sein muss, spielen in dieser Arbeit auch Klammersprachen ein Rolle. Wir arbeiten hier mit kontextfreien Klammersprachen, die von kontextfreien Klammergrammatiken erzeugt werden. Eine Grammatik ist kontextfrei, wenn sie ein 4-Tupel (N, T, P, S)

ist, wobei N die Menge der Nichtterminalsymbole, T die Menge der Terminalsymbole, P die Menge der Regeln und S das Startsymbol ist. Dabei gilt, dass auf der linken Seite jeder Regel genau ein Nichtterminalsymbol steht. Weiter ist eine Grammatik eine Klammergrammatik, wenn die rechte Seite jeder Regel von Klammersymbolen umschlossen ist, die selbst aus T sind. Setzt man bei den Formeln der Aussagenlogik eine komplette Klammerung inklusive der äußeren Klammern voraus, so kann die Menge aller aussagenlogischen Formeln, die keine Variablen enthalten, durch solch eine Klammergrammatik beschrieben werden.

Sowohl in der Modallogik als auch in der intuitionistischen Logik sind Graphen wesentliche Bestandteile der Modelle. Ein Graph G ist ein Paar (V, E) wobei V die Menge der Knoten und $E \subseteq V \times V$ die Menge der Kanten ist. Da wir komplexitätstheoretische Untersuchungen durchführen und das Berechnungsmodell der Turingmaschine zu Grunde legen, müssen wir Graphen auch für Turingmaschinen lesbar machen. Daher kodieren wir einen Graphen durch eine Adjazenzmatrix. Das ist eine Matrix, in der für jedes Knotenpaar gespeichert wird, ob zwischen den beiden Knoten eine Kante existiert oder nicht. In diesem Sinne ist die Größe eines Graphen immer quadratisch in der Anzahl seiner Knoten – unabhängig von der tatsächlichen Anzahl der Kanten.

Die Kanten eines Graphen bilden eine Relation über der Menge der Knoten. Diese Relation kann bestimmte Eigenschaften haben, wir sagen dann, dass der Graph diese Eigenschaften hat. Sei $G = (V, E)$ ein Graph, dann ist G

irreflexiv	$\iff$	$\forall v \in V : (v, v) \notin E$,
reflexiv	$\iff$	$\forall v \in V : (v, v) \in E$,
transitiv	$\iff$	$\forall u, v, w \in V :$ wenn $(u, v), (v, w) \in E,$ dann $(u, w) \in E$,
antisymmetrisch	$\iff$	$\forall v, w \in V :$ wenn $(v, w), (w, v) \in E,$ dann $v = w$,
ungerichtet	$\iff$	$\forall v, w \in V :$ wenn $(v, w) \in E,$ dann $(w, v) \in E$,
linear	$\iff$	$\forall v, w \in V : (v, w) \in E$ oder $(w, v) \in E,$ oder $v = w$.

Wir bezeichnen G als Halbordnung, wenn G reflexiv und transitiv ist und als gerichtete Halbordnung, wenn G eine Halbordnung ist und es für je zwei Knoten $v, w \in V$ einen dritten Knoten $u \in V$ gibt, so dass $(v, u) \in E$ und $(w, u) \in E$ gilt. Weiter ist G eine lineare Ordnung, wenn G transitiv und linear ist.

Dieses einführende Kapitel ist wie folgt gegliedert: In Abschnitt 2.1 führen wir die Logik ein. Die intuitionistische Logik wird als eine spezielle Modallogik definiert, deswegen beginnt dieser Abschnitt mit der Vorstellung der Modallogik. Die für uns wesentlichen Teile der Komplexitätstheorie behandeln wir in Abschnitt 2.2. Dort führen wir die notwendigen Komplexitätsklassen ein und gehen in einem Unterabschnitt auf vollständige Probleme ein, die in dieser Arbeit als Hilfsmittel verwendet werden. In Abschnitt 2.3 definieren wir im ersten Teil die Entscheidungsprobleme, die später analysiert werden. Im zweiten Teil geben wir eine unvollständige Auswahl von bereits bekannten und einfach zu zeigenden Resultaten an.

2.1 Logik

In diesem Abschnitt wollen wir die Logiken dieser Arbeit vorstellen. Dabei geht es um intuitionistische Logiken und Modallogiken. Zuerst führen wir in Abschnitt 2.1.1 die Modallogiken ein. Dann definieren wir die intuitionistischen Logiken als spezielle Modallogiken in Abschnitt 2.1.2.
Die wesentlichen Bestandteile der Einführung einer Logik sind die Syntax (Wie sehen die Formeln aus?) und die Semantik (Welche Modelle werden zur Interpretation verwendet und wie wird interpretiert?). Wir betrachten in dieser Arbeit eine Logik immer als ein Paar, bestehend aus einer Menge von Formeln und einer Menge von Modellen. Dass dieser Logikbegriff zur üblichen Definition über Mengen von Theoremen passend ist, wird in Abschnitt 2.1.1 ebenfalls geklärt.
Im abschließenden Abschnitt 2.1.3 gehen wir noch kurz auf Heyting-Algebren ein, da diese neben den Kripke-Modellen ebenfalls als Semantik für die intuitionistischen Logiken verwendet werden können. Weil man über den Heyting-Algebren ein Formelauswertungsproblem, wie wir es betrachten, nicht definieren kann, spielen sie für uns als Semantik eine untergeordnete Rolle. Wir verwenden sie aber als technische Hilfsmittel in späteren Beweisen.

2.1.1 Modallogik

Jetzt geht es um Modallogik. Wir geben zuerst die Syntax an, führen dann die Kripke-Semantik ein und definieren anschließend den Begriff der Modallogik formal. Abschließend geben wir einige Modallogiken an. Eine ausführliche Einführung in die Modallogik kann [BdV01] und [CZ97] entnommen werden.

Syntax

Die Grundlage jeder Logik bildet eine Sprache. Alle Worte dieser Sprache sind Formeln der Logik. Die Sprache $\mathfrak{F}$ der Modallogik ist eine Erweiterung der Sprache der Aussagenlogik – es gibt einen zusätzlichen einstelligen Operator.

Definition 2.1 *Sei* VAR *eine abzählbare Menge von Variablen. Die Sprache* $\mathfrak{F}$ *ist die Menge aller Formeln der Form*

$$\alpha \quad ::= \quad p \mid \bot \mid (\alpha \to \alpha) \mid \Box\alpha,$$

wobei $p \in \text{VAR}$.

Durch Kombination der Operatoren aus Definition 2.1 lassen sich neue Operatoren definieren. Wir nutzen in dieser Arbeit folgende abkürzende Schreibweisen:

$$\begin{aligned} \neg\alpha &:= \alpha \to \bot\ , \\ \neg\neg\alpha &:= \neg(\neg\alpha)\ , \\ \top &:= \neg\bot\ , \end{aligned}$$

$$\begin{aligned}
\alpha \vee \beta &:= (\neg\alpha) \to \beta \ , \\
\alpha \wedge \beta &:= \neg((\neg\alpha) \vee (\neg\beta)) \ , \\
\alpha \leftrightarrow \beta &:= (\alpha \to \beta) \wedge (\beta \to \alpha) \ , \\
\Diamond\alpha &:= \neg(\Box(\neg\alpha)) \ .
\end{aligned}$$

Die Elemente aus VAR werden auch als *atomare Aussagen* bezeichnet. Die nullstelligen Operatoren $\bot$ und $\top$ bezeichnen wir als *Konstanten*. Die einstelligen Operatoren $\Diamond$ und $\Box$ nennen wir *modale Operatoren*. Die Operatoren $\wedge$, $\vee$, $\to$ und $\neg$ bezeichnen wir als *aussagenlogische Operatoren*. Die folgenden Operatoren werden später vornehmlich in der intuitionistischen Logik verwendet:

$$\begin{aligned}
\alpha \rightarrowtriangle \beta &:= \Box(\alpha \to \beta) \ , \\
\neg\!\!\!\rightarrowtriangle \alpha &:= \alpha \rightarrowtriangle \bot \ , \\
\neg\!\!\!\rightarrowtriangle\neg\!\!\!\rightarrowtriangle \alpha &:= \neg\!\!\!\rightarrowtriangle(\neg\!\!\!\rightarrowtriangle\alpha) \ , \\
\alpha \leftrightarrowtriangle \beta &:= (\alpha \rightarrowtriangle \beta) \wedge (\beta \rightarrowtriangle \alpha) \ .
\end{aligned}$$

Der Operator $\rightarrowtriangle$ wird (im Kontext der Modallogik) als *strikte* oder (im Kontext intuitionistischer Logiken) als *intuitionistische Implikation* bezeichnet, $\neg\!\!\!\rightarrowtriangle$ ist die *strikte* oder *intuitionistische Negation*. Ist ein Operator zweistellig, schreiben wir ihn zwischen die Formeln. Operatoren mit mehr als zwei Stellen spielen in dieser Arbeit eine untergeordnete Rolle.[1] Um eine bessere Lesbarkeit der Formeln zu ermöglichen, setzen wir folgende Bindungsstärken voraus: Die aussagenlogischen Operatoren binden stärker als die modalen Operatoren. Grundsätzlich bindet $\neg$ am stärksten und $\wedge$ und $\vee$ binden stärker als $\to$. Die strikte Implikation hat dieselbe Bindungsstärke wie die normale Implikation. Im Sinne dieser Bindungsstärken können Klammen gespart werden. Ebenso werden äußere Klammern weggelassen. Zum Beispiel schreiben wir $\Box(a \to b \wedge c)$ statt $(\Box(a \to (b \wedge c)))$.
In dieser Arbeit werden oft nur Teilmengen von $\mathfrak{F}$ betrachtet. Dabei beschränken wir einerseits die Operatoren und andererseits die Anzahl der Variablen. Ein *Formelfragment* ist eine Teilmenge von $\mathfrak{F}$, die unter Einsetzung abgeschlossen ist.

Definition 2.2 *Es seien $n \in \mathbb{N} \cup \{\infty\}$, $k > 0$ und $O_1^{s_1}, O_2^{s_2}, \ldots, O_k^{s_k}$ Operatoren mit den Stelligkeiten $s_1, s_2, \ldots, s_k \geq 0$. $\mathrm{VAR}_n \subseteq \mathrm{VAR}$ enthalte genau n Elemente, für $n = \infty$ gilt $\mathrm{VAR}_n = \mathrm{VAR}$ und für $n = 0$ gilt $\mathrm{VAR}_n = \varnothing$. Die Menge $\mathcal{F} = \mathfrak{F}[O_1^{s_1}, O_2^{s_2}, \ldots, O_k^{s_k}, n] \subseteq \mathfrak{F}$ ist ein Formelfragment von $\mathfrak{F}$, wenn Folgendes gilt:*

(1) $\mathrm{VAR}_n \subseteq \mathcal{F}$.

(2) *Für $1 \leq i \leq k$ und $\alpha_1, \alpha_2, \ldots, \alpha_{s_i} \in \mathcal{F}$ gilt: $(O_i^{s_i}(\alpha_1, \alpha_2, \ldots, \alpha_{s_i})) \in \mathcal{F}$.*

(3) *$\mathcal{F}$ enthält keine weiteren Formeln.*

[1] Ist ein Operator O^n n-stellig, schreibt man ihn vor die n Formeln: $(O^n(\alpha_1, \alpha_2, \ldots, \alpha_n))$.

Aus Definition 2.2 kann man sofort $\mathfrak{F} = \mathfrak{F}[\bot, \to, \Box, \infty]$ ableiten. Zur Vereinfachung lassen wir ∞ in dieser Schreibweise grundsätzlich weg. Ebenso schreiben wir $\mathfrak{F}[n]$ statt $\mathfrak{F}[\bot, \to, \Box, n]$, da $\bot$, $\to$ und $\Box$ die Operatoren aus Definition 2.1 sind. Wir gehen in dieser Arbeit davon aus, dass $\mathrm{VAR}_n = \{p_1, p_2, \ldots, p_n\}$ ist, wenn die Variablen nicht anders benannt wurden, da die konkrete Bezeichnung der Variablen nicht relevant ist.

Semantik

Nun sollen die eben definierten Formeln interpretiert werden. In der Aussagenlogik verwendet man dafür Belegungen, welche den Variablen die Wahrheitswerte `true` und `false` zuordnen. In der Modallogik ist die Interpretation etwas komplizierter, da es zusätzlich den $\Box$-Operator gibt.
Wir verwenden in dieser Arbeit *Kripke-Modelle* [Kri63a] als Grundlage für die Interpretation von Formeln. Ein Kripke-Modell besteht aus einer Menge von *Welten*, einer *Sichtbarkeitsrelation*, welche die Welten verbindet und einer *Belegungsfunktion*, mit der den Welten atomare Aussagen zugeordnet werden, die in ihnen erfüllt sind. Die Menge der Welten und die Sichtbarkeitsrelation bilden den *Rahmen* des Modells. Im Gegensatz zu den Belegungen in der Aussagenlogik hat jede Welt in einem Kripke-Modell eine eigene Belegung. In diesem Sinne kann man die durch Kripke-Modelle beschriebene Kripke-Semantik als Verallgemeinerung oder Erweiterung der Semantik der Aussagenlogik sehen. Die Belegung der Variablen in einer (Teil)Formel hängt von der konkreten Welt ab, in der man sie interpretiert. Mit dem $\Box$-Operator kann man zwischen verschiedenen Welten eines Modells navigieren. Da wir komplexitätstheoretische Untersuchungen durchführen und die Modelle oft auch Teil der Eingabe sind, betrachteten wir nur endliche Modelle.

Definition 2.3 *Es seien W eine endliche Menge und $S \subseteq W \times W$ eine Relation über W. Weiter seien $V \subseteq \mathrm{VAR}$ und $\xi : V \to \mathfrak{P}(W)$ eine Funktion. Dann ist das Quadrupel $\mathcal{M} = (W, S, \xi, V)$ ein Kripke-Modell und das Paar (W, S) der zugrundeliegende Rahmen. Die Elemente von W heißen Welten, S ist die Sichtbarkeitsrelation und ξ die Belegungsfunktion. Mit $\mathfrak{K}$ bezeichnen wir die Menge aller Kripke-Modelle.*

Um über die Verhältnisse zwischen den Welten anschaulich sprechen zu können, führen wir weitere Begriffe ein. Es seien $\mathcal{M} = (W, S, \xi, V)$ ein Kripke-Modell und $w \in W$ eine Welt aus $\mathcal{M}$. Für $v \in W$ mit $(w, v) \in S$ sagen wir, dass die Welt w die Welt v *sieht* und bezeichnen v als *Nachfolger* von w. Alle Nachfolger v von w mit $w \neq v$ sind *echte Nachfolger* von w. Wir bezeichnen die Welt w als *maximal*, wenn sie keine echten Nachfolger hat. Die Begriffe *Vorgänger*, *echter Vorgänger* und *minimal* werden analog verwendet. Des Weiteren führen wir folgende abkürzende Schreibweisen ein:

$$W_{w\Uparrow} := \{v \in W \mid (w, v) \in S\} \quad \text{(Menge aller Nachfolger von } w\text{)},$$
$$W_{w\uparrow} := W_{w\Uparrow} \setminus \{w\} \quad \text{(Menge aller echten Nachfolger von } w\text{)}.$$

Im Folgenden sprechen wir oftmals nur von *Modellen* und meinen damit aber grundsätzlich Kripke-Modelle.
Für die Formelmenge $\mathfrak{F}$ haben wir über den Begriff der Formelfragmente „sinnvolle“ Teilmengen definiert (Definition 2.2). Für die Menge $\mathfrak{K}$ aller Modelle wollen wir ebenfalls „sinnvolle“ Teilmengen – die sogenannten *Modellklassen* – definieren. In dieser Arbeit betrachten wir Modelle, bei denen die Sichtbarkeitsrelation eingeschränkt ist oder die Belegungsfunktion zusätzliche Forderungen erfüllt.

Definition 2.4 *Sei $\mathcal{K} \subseteq \mathfrak{K}$ eine Menge von Modellen. Wir bezeichnen $\mathcal{K}$ als Modellklasse, wenn es eine eine prädikatenlogische Formel P mit Gleichheit und ohne freie Variablen gibt, so dass für jedes Modell $\mathcal{M}$ aus $\mathfrak{K}$ gilt, dass P unter $\mathcal{M}$ genau dann wahr wird, wenn $\mathcal{M}$ ein Modell aus $\mathcal{K}$ ist.*

Diese Definition wird auch durch die Korrespondenztheorie motiviert. Betrachtet man Logiken im klassischen syntaktischen Sinne zum Beispiel als Mengen von Theoremen, die mit Axiomen und Ableitungsregeln bewiesen werden, so gibt es für viele Axiome einen Zusammenhang zwischen ihrer (intuitiven) Aussage und der Eigenschaft von Modellen. Zum Beispiel korrespondiert das Axiom $\Diamond\Diamond p \to \Diamond p$ mit der prädikatenlogischen Formel $\forall u, v, w \in W : (((u, v) \in S \wedge (v, w) \in S) \to (u, w) \in S)$. Das bedeutet, in einer Modallogik, in der $\Diamond\Diamond p \to \Diamond p$ und alle durch uniforme Ersetzung entstehenden Formeln immer wahr (also Tautologien) sind, müssen die Modelle die Eigenschaft $\forall u, v, w \in W : (((u, v) \in S \wedge (v, w) \in S) \to (u, w) \in S)$ erfüllen – also transitiv sein. Vertieft werden kann diese Einführung in [BdV01] und [Lad77], eine übersichtliche Aufstellung von Axiomen und ihren korrespondierenden Eigenschaften gibt es in [Sch02].
Durch eine *Interpretation* ist es möglich, Formeln in Welten von Modellen auszuwerten – Formeln können in einer Welt erfüllt sein oder nicht. Dafür definieren wir die Relation $\models$, die oft auch *Erfüllt-sein-Relation* genannt wird.

Definition 2.5 *Seien $\mathcal{M} = (W, S, \xi, V)$ ein Modell, $w \in W$ eine Welt und $\alpha, \beta \in \mathfrak{F}$ Formeln. Wir definieren $\models$ wie folgt:*

(1) $\mathcal{M}, w \not\models \bot$,

(2) $\mathcal{M}, w \models p \iff w \in \xi(p),\ p \in V$,

(3) $\mathcal{M}, w \models \alpha \to \beta \iff$ *wenn* $\mathcal{M}, w \models \alpha$, *dann* $\mathcal{M}, w \models \beta$,

(4) $\mathcal{M}, w \models \Box\alpha \iff \forall v \in W, (w, v) \in S : \mathcal{M}, v \models \alpha$.

Für ein Modell $\mathcal{M} \in \mathfrak{K}$, eine Welt w aus diesem Modell und eine Formel $\alpha \in \mathfrak{F}$ sagen wir, dass α von w in $\mathcal{M}$ *erfüllt* wird, wenn $\mathcal{M}, w \models \alpha$ gilt. Wird α von jedem Modell einer Modellklasse $\mathcal{K}$ erfüllt, nennen wir α *$\mathcal{K}$-gültig* oder eine *$\mathcal{K}$-Tautologie*. Gibt es für eine Formel α ein Modell $\mathcal{M} \in \mathcal{K}$ mit einer Welt, in der sie erfüllt wird, so bezeichnen wir α als $\mathcal{K}$-erfüllbar. Gibt es in keinem Modell aus $\mathcal{K}$ eine solche Welt, dann ist α $\mathcal{K}$-unerfüllbar.

Modallogiken

Jetzt können wir den Begriff der Modallogik, wie wir ihn im weiteren Verlauf verwenden werden, definieren.

Definition 2.6 *Seien $\mathcal{F} \subseteq \mathfrak{F}$ ein Formelfragment und $\mathcal{K} \subseteq \mathfrak{K}$ eine Modellklasse, dann ist das Paar $(\mathcal{F}, \mathcal{K})$ eine Modallogik. Für die Modallogik $\mathrm{L} = (\mathcal{F}, \mathcal{K})$ bezeichnen wir jede Formel aus $\mathcal{F}$ als* L*-Formel und jedes Modell aus $\mathcal{K}$ als* L*-Modell.*

Wie bereits erwähnt, werden in der Literatur Logiken oftmals als eine Menge von Theoremen charakterisiert, die sich unter Anwendung bestimmter Regeln beweisen lassen.[2] Diese Charakterisierung betrachtet Logiken von der syntaktischen Seite.[3] Beweisverfahren für Modallogiken sind beispielsweise das natürliche Schließen (siehe u.a. [Ind10]) und das Tableauverfahren (siehe u.a. [NS97]). Da wir uns in dieser Arbeit für die semantische Seite interessieren, fassen wir Modallogiken als Paare, bestehend aus einer Menge von Formeln und einer Menge von Modellen, auf. Dem Paar $(\mathcal{F}, \mathcal{K})$, das in Definition 2.6 Modallogik genannt wird, kann die Menge aller $\mathcal{K}$-Tautologien (oder aller $\mathcal{K}$-unerfüllbaren Formeln) aus $\mathcal{F}$ eindeutig zugeordnet werden. In diesem Sinne besteht zwischen dem Logikbegriff aus Definition 2.6 und der Definition über Mengen von syntaktisch beweisbaren Theoremen (für korrekte und vollständige Systeme) kein Unterschied. Natürlich kann es für eine Menge von Formeln, die Theoreme sein sollen, mehrere Logiken nach unserer Definition geben.
Mit Hilfe der Formelfragmente und der Modellklassen können wir Fragmente von Logiken angeben. Sei $\mathcal{F}[O, n]$ ein Formelfragment[4] gemäß Definition 2.2 von $\mathfrak{F}$ und $\mathcal{K}[n]$ eine Modellklasse, bei der die Variablenmenge in allen Modellen höchstens n-elementig ist. Dann ist die Logik $\mathrm{L}[O, n] := (\mathcal{F}[O, n], \mathcal{K}[n])$ ein Fragment von $\mathrm{L} = (\mathcal{F}, \mathcal{K})$ (und jeder Logik $(\mathcal{F}', \mathcal{K}')$ mit $\mathcal{F} \subseteq \mathcal{F}'$ und $\mathcal{K} \subseteq \mathcal{K}'$).
Jetzt werden noch zwei Äquivalenzbegriffe definiert. Zum einen können Formeln in Bezug auf eine Modellklasse äquivalent sein und zum anderen können Welten bezüglich eines Formelfragments äquivalent sein.

Definition 2.7 *Es seien $\mathcal{K}$ eine Modellklasse und $\mathcal{F}$ ein Formelfragment.*

(1) *Seien $\mathcal{M}_1, \mathcal{M}_2 \in \mathcal{K}$, w_1 eine Welt aus $\mathcal{M}_1$ und w_2 eine Welt aus $\mathcal{M}_2$. Wir sagen, $(\mathcal{M}_1, w_1)$ ist $\mathcal{F}$-äquivalent zu $(\mathcal{M}_2, w_2)$, wenn für alle Formeln $\alpha \in \mathcal{F}$*

$$\mathcal{M}_1, w_1 \models \alpha \iff \mathcal{M}_2, w_2 \models \alpha$$

gilt und schreiben $(\mathcal{M}_1, w_1) \equiv_{\mathcal{F}} (\mathcal{M}_2, w_2)$.

[2] Je nach Kalkül kann es zusätzlich Axiome geben oder nicht.
[3] Es gibt Kalküle, in denen die unerfüllbaren Formeln bewiesen werden (das Tableauverfahren) und welche, in denen die Tautologien bewiesen werden (natürliches Schließen). Solche Kalküle sind sinnvoll definiert, wenn sie korrekt und vollständig sind.
[4] O bezeichnet in diesem Fall eine Menge von Operatoren.

(2) *Seien $\alpha_1, \alpha_2 \in \mathcal{F}$. Wir sagen, α_1 ist $\mathcal{K}$-äquivalent zu α_2, wenn für alle Modelle $\mathcal{M}$ aus $\mathcal{K}$ und alle Welten w aus $\mathcal{M}$*

$$\mathcal{M}, w \models \alpha_1 \iff \mathcal{M}, w \models \alpha_2$$

gilt und schreiben $\alpha_1 \equiv_{\mathcal{K}} \alpha_2$.

Mit $[\alpha]$ bezeichnen wir die Menge aller Formeln aus $\mathcal{F}$, die $\mathcal{K}$-äquivalent zu α sind. Wie in der Aussagenlogik gilt auch hier, dass zwei Formeln α_1 und α_2 $\mathcal{K}$-äquivalent sind, wenn $\alpha_1 \leftrightarrow \alpha_2$ eine $\mathcal{K}$-Tautologie ist.

Einige Vertreter

Die allgemeinste Modallogik ist K = $(\mathfrak{F}, \mathfrak{K})$ – jedes Modell aus Definition 2.1 ist zugelassen. Eine häufige Forderung an die Modelle ist die Transitivität. Bei der Logik K4 = $(\mathfrak{F}, \mathfrak{K}_{trans})$ sind nur Modelle basierend auf transitiven Rahmen zugelassen. Dabei ist $\mathfrak{K}_{trans}$ die Menge aller Modelle, deren Rahmen transitiv sind. Syntaktisch gesehen entsteht K4 aus K durch Hinzunahme des Axioms $\Diamond\Diamond p \rightarrow \Diamond p$ mit Abschluss unter uniformer Ersetzung, Modus Ponens und der Notwendigkeitsregel. Eine weitere Logik ist S4 = $(\mathfrak{F}, \mathfrak{K}_{HO})$. Die Rahmen der Modelle sind hier transitiv und reflexiv (also eine Halbordnung). Syntaktisch sind hier zu K die Axiome $\Diamond\Diamond p \rightarrow \Diamond p$ für die Transitivität und $p \rightarrow \Diamond p$ für die Reflexivität hinzugekommen. Die irreflexive Variante ist die Logik PrL = $(\mathfrak{F}, \mathfrak{K}_{irr})$, bei der alle Modelle einen transitiven und irreflexiven Rahmen haben. Weitere Logiken sind S4.2 = $(\mathfrak{F}, \mathfrak{K}_{gerHO})$, deren Sichtbarkeitsrelation gerichtete Halbordnungen sind und S4.3 = $(\mathfrak{F}, \mathfrak{K}_{linO})$ mit linearen Ordnungen als Sichtbarkeitsrelation. Eine Übersicht über die Logiken und ihre Modelleigenschaften ist in Tabelle 2.1 zu sehen.

Logik	Modelleigenschaften
K	keine Einschränkungen
K4	transitiv
PrL	transitiv und irreflexiv
S4	transitiv und reflexiv (= Halbordnung)
S4.2	gerichtete Halbordnung
S4.3	lineare Ordnung

Tabelle 2.1: Modallogiken und ihre Modelleigenschaften.

2.1.2 Intuitionistische Logik

In diesem Abschnitt führen wir die intuitionistische Logik ein. Die intuitionistischen Logiken sind im Sinne unserer Einführung spezielle Modallogiken. Syntax

und Semantik basieren auf den Definitionen 2.1 und 2.3, der entscheidende Unterschied zur Modallogik ist die Monotonie. Monotonie besagt im Wesentlichen, dass eine Formel, die einmal in einer Welt erfüllt ist, auch in allen Nachfolgern dieser Welt erfüllt sein soll. Außerdem müssen die Modelle kreisfrei und transitiv sein.
Wir definieren wieder zuerst die Syntax, dann die Semantik, geben im Anschluss einige Vertreter an und zeigen schließlich den Zusammenhang zwischen diesen Vertretern und bestimmten Modallogiken. Die Grundlagen der intuitionistischen Logik sind im Wesentlichen aus [Gab81] und [vD04] entnommen und können dort weiter vertieft werden.

Syntax

Die Sprache der intuitionistischen Logik ist eine Teilmenge von $\mathfrak{F}$, wobei statt der normalen Implikation die intuitionistische Implikation verwendet wird.

Definition 2.8 *Die Menge* $\mathfrak{F}^{\mathsf{i}} = \mathfrak{F}[\bot, \wedge, \vee, \rightarrowtriangle]$ *ist die Menge aller intuitionistischen Formeln.*

Da die intuitionistische Implikation $\rightarrowtriangle$ anders interpretiert wird als die normale Implikation, lassen sich $\wedge$ und $\vee$ nicht durch $\rightarrowtriangle$ und $\bot$ darstellen.[5] Des Weiteren gibt es in der intuitionistischen Logik keine normale Negation $\neg$, sondern nur die intuitionistische Negation $\neg\alpha = \alpha \rightarrowtriangle \bot$.
Ein Formelfragment von $\mathfrak{F}$ ist ein intuitionistisches Formelfragment, wenn es nur Formeln mit Operatoren aus $\{\wedge, \vee, \rightarrowtriangle, \bot\}$ enthält. Außerdem sind natürlich auch Operatoren erlaubt, die sich aus $\wedge$, $\vee$, $\rightarrowtriangle$ und $\bot$ konstruieren lassen – wie zum Beispiel die intuitionistische Negation. Diese intuitionistischen Formelfragmente werden auch als Formelfragmente von $\mathfrak{F}^{\mathsf{i}}$ bezeichnet. Zur Verdeutlichung schreiben wir in diesen Fällen das hochgestellte i an die Formelmenge. Wenn eindeutig klar ist, dass es sich um eine intuitionistische Logik handelt, lassen wir – ähnlich wie bei den Formelfragmenten der Modallogiken – die Standardoperatoren $\bot$, $\wedge$, $\vee$ und $\rightarrowtriangle$ weg, wenn alle vorkommen. Wir schreiben also beispielsweise $\mathfrak{F}^{\mathsf{i}}[n]$ statt $\mathfrak{F}^{\mathsf{i}}[\bot, \wedge, \vee, \rightarrowtriangle, n]$.

Semantik

Als Semantik für die intuitionistischen Logiken verwenden wir ebenfalls Kripke-Modelle [Kri63b, Kri65]. Gegenüber den in Definition 2.3 definierten Modellen haben die für die intuitionistische Logik weitere Einschränkungen.

Definition 2.9 *Sei* $\mathcal{M} = (W, S, \xi, V)$ *ein Kripke-Modell. Zusätzlich sei* (W, S) *kreisfrei und transitiv. Dann ist* $\mathcal{M}$ *monoton, wenn* $\xi^{-1}(w_1) \subseteq \xi^{-1}(w_2)$ *für alle* $w_1, w_2 \in W$ *mit* $(w_1, w_2) \in S$ *gilt. Ist ein Modell monoton, dann bezeichnen wir es als intuitionistisches Kripke-Modell.* $\mathfrak{K}^{\mathsf{i}}$ *ist die Menge aller intuitionistischen Kripke-Modelle.*

[5] In der intuitionistischen Logik gelten die De Morgan'schen Gesetze nicht uneingeschränkt.

Intuitionistische Modellklassen sind Modellklassen, in denen jedes Modell ein intuitionistisches Kripke-Modell ist.
Die Interpretation der intuitionistischen Formeln über den intuitionistischen Modellen folgt direkt aus der Interpretation für die Modallogiken (Definition 2.5). Da die intuitionistische Implikation $\rightarrowtriangle$ eine zentrale Rolle spielt, geben wir ihre Interpretation hier noch einmal explizit an. Ein Beispiel ist in Abbildung 2.1 zu sehen.

Bemerkung 2.10 *Es seien $\mathcal{M} = (W, S, \xi, V)$ und $\mathcal{M} \in \mathfrak{K}^{\mathsf{i}}$ ein intuitionistisches Kripke-Modell, $w \in W$ eine Welt und $\alpha, \beta \in \mathfrak{F}^{\mathsf{i}}$ Formeln. Dann gilt*

(1) $\mathcal{M}, w \models \alpha \rightarrowtriangle \beta \iff \forall v \in W, (w, v) \in S : \mathcal{M}, v \models \alpha \Rightarrow \mathcal{M}, v \models \beta$,

(2) $\mathcal{M}, w \models \neg\alpha \iff \forall v \in W, (w, v) \in S : \mathcal{M}, v \not\models \alpha$,

(3) $\mathcal{M}, w \models \neg\neg\alpha \iff \forall v \in W, (w, v) \in S : \exists u \in W, (v, u) \in S : \mathcal{M}, u \models \alpha$.

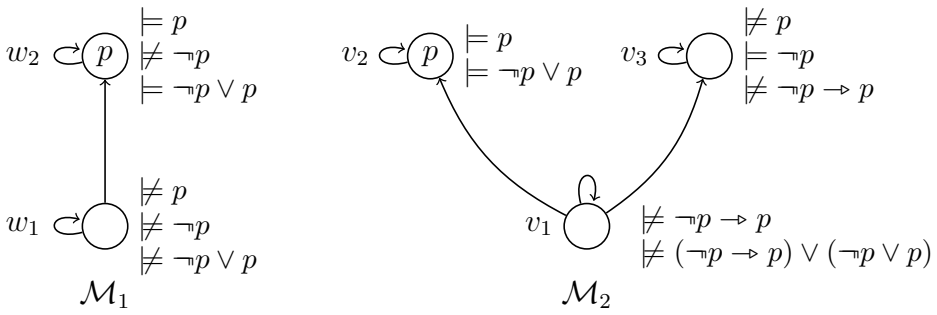

Abbildung 2.1: Das Modell $\mathcal{M}_1$ (links) ist das Standardbeispiel, in dem gezeigt wird, dass das starke Gesetz des ausgeschlossenen Dritten $p \vee \neg p$ in der intuitionistischen Logik nicht gültig ist. Das Modell $\mathcal{M}_2$ (rechts) zeigt, wie die intuitionistische Implikation funktioniert.

In Abbildung 2.1 wird der Unterschied zwischen intuitionistischer und normaler Implikation noch einmal deutlich: Während man bei der Auswertung der normalen Implikation nur die Welt betrachtet, in der man auswertet, werden bei der intuitionistischen Implikation alle Nachfolger dieser Welt angeschaut. Die intuitionistische Negation wird ebenfalls so interpretiert. Damit ist die Monotonie sichergestellt. Sie charakterisiert bei der semantischen Betrachtung maßgeblich den Unterschied zwischen den allgemeinen Modallogiken und intuitionistischen Logiken. Zum einen wird gefordert, dass die Belegungsfunktion in allen intuitionistischen Kripke-Modellen monoton ist (Definition 2.9) und zum anderen ist auch die Interpretation der intuitionistischen Implikation so definiert (Bemerkung 2.10). Für $\wedge$ und $\vee$ gilt die Monotonie dann ebenfalls.

Lemma 2.11 *Seien $\mathcal{M} = (W, S, \xi, V)$ und $\mathcal{M} \in \mathfrak{K}^{\mathsf{i}}$ ein intuitionistisches Kripke-Modell, $w \in W$ eine Welt, und $\alpha \in \mathfrak{F}^{\mathsf{i}}$ eine Formel. Dann folgt aus $\mathcal{M}, w \models \alpha$ auch $\mathcal{M}, v \models \alpha$ für alle $v \in W$ mit $(w, v) \in S$.*

Dieses Lemma lässt sich einfach mit vollständiger Induktion über Aufbau von α zeigen.

Einige Vertreter

Jetzt wollen wir einige Logiken angeben, die aus der Literatur bekannt sind und für die in dieser Arbeit das Formelauswertungsproblem untersucht wird. Bei diesen Logiken wird zuerst nur die Modellklasse eingeschränkt. Dies geschieht über zusätzliche Bedingungen an die Sichtbarkeitsrelation. Später beschränken wir auch die Verwendung von Operatoren und die Zahl der vorkommenden Variablen.
Die allgemeinste intuitionistische Logik ist BPL, die sogenannte *Basic Propositional Logic*. Sie wurde 1980 von Visser [Vis80] eingeführt. Nach unserer Definition 2.6 ist BPL die Logik $(\mathfrak{F}^{\mathsf{i}}, \mathfrak{K}^{\mathsf{i}})$, es gibt also keine weiteren Anforderungen an die Modellklasse.[6] Am weitesten verbreitet und am meisten untersucht ist die Logik IPL, *Intuitionistic Propositional Logic*. Hier wird zusätzlich gefordert, dass die Sichtbarkeitsrelation aller Modelle reflexiv ist. Im Weiteren bezeichnen wir die Klasse der reflexiven intuitionistischen Modelle mit $\mathfrak{K}^{\mathsf{i}}_{\mathit{refl}}$. IPL ist also $(\mathfrak{F}^{\mathsf{i}}, \mathfrak{K}^{\mathsf{i}}_{\mathit{refl}})$. Ebenfalls von Visser [Vis80] wurde 1980 FPL (*Formal Propositional Logic*) eingeführt. In dieser Logik wird ausschließlich über irreflexiven Modellen interpretiert. Wir bezeichnen die Klasse aller irreflexiven intuitionistischen Modelle mit $\mathfrak{K}^{\mathsf{i}}_{\mathit{irr}}$, FPL ist demnach $(\mathfrak{F}^{\mathsf{i}}, \mathfrak{K}^{\mathsf{i}}_{\mathit{irr}})$. Weitere Logiken sind KC [DL59], auch *Jankov's Logic* oder *De Morgan Logic* genannt und LC [Dum59], auch *Gödel-Dummett Logic* genannt. Bei KC ist die Sichtbarkeitsrelation der Modelle eine gerichtete Halbordnung, das heißt, sie ist reflexiv und je zwei Welten haben einen gemeinsamen Nachfolger. In diesem Sinne ist KC $= (\mathfrak{F}^{\mathsf{i}}, \mathfrak{K}^{\mathsf{i}}_{\mathit{gerHO}})$, wobei $\mathfrak{K}^{\mathsf{i}}_{\mathit{gerHO}} \subseteq \mathfrak{K}^{\mathsf{i}}_{\mathit{refl}}$ die Menge aller Modelle ist, deren Sichtbarkeitsrelation eine gerichtete Halbordnung ist. Jedes KC-Modell hat genau eine maximale Welt, da wir nur endliche Modelle betrachten. Die Sichtbarkeitsrelation bei den Modellen aus LC ist eine lineare Ordnung. Für zwei Welten gilt also immer, dass die eine Nachfolger der anderen ist oder umgekehrt. Es gilt LC $= (\mathfrak{F}^{\mathsf{i}}, \mathfrak{K}^{\mathsf{i}}_{\mathit{linO}})$, wobei $\mathfrak{K}^{\mathsf{i}}_{\mathit{linO}} \subseteq \mathfrak{K}^{\mathsf{i}}_{\mathit{refl}}$ die Menge aller Modelle ist, deren Sichtbarkeitsrelation eine lineare Ordnung ist. Selbst die Aussagenlogik (AL) kann als intuitionistische Logik aufgefasst werden. Ist $\mathfrak{K}^{\mathsf{i}}_1$ die Klasse aller Modelle, die nur aus genau einer Welt bestehen und deren Sichtbarkeitsrelation reflexiv ist, so ist die Menge der $\mathfrak{K}^{\mathsf{i}}_1$-Tautologien aus $\mathfrak{F}^{\mathsf{i}}$ genau die Menge der aussagenlogischen Tautologien. Man kann also die Aussagenlogik auch durch $(\mathfrak{F}^{\mathsf{i}}, \{(\{w\}, \{(w, w)\}, \xi, V) \mid w,\ \xi \text{ und } V \subseteq \mathrm{VAR} \text{ beliebig}\})$ charakterisieren. Insbesondere gilt bei der Interpretation von Formeln aus $\mathfrak{F}^{\mathsf{i}}$ über Modellen mit nur einer sich selbst sehenden Welt, dass $\rightarrowtriangle$ und $\rightarrow$ dieselbe Bedeutung haben.

[6] Bei intuitionistischen Logiken gehen wir grundsätzlich davon aus, dass die Modelle transitiv, kreisfrei und monoton sind.

Dies überträgt sich auch auf $\neg$ und $\neg$. Eine Übersicht über diese Logiken und die Eigenschaften ihrer Modelle ist in Tabelle 2.2 gegeben.

Logik	Modelleigenschaften
BPL	transitiv
FPL	transitiv und irreflexiv
IPL	transitiv und reflexiv (= Halbordnung)
KC	gerichtete Halbordnung
LC	lineare Ordnung
AL	reflexiv und genau eine Welt

Tabelle 2.2: Intuitionistische Logiken und ihre Modelleigenschaften.

Die Logiken AL, LC und KC sind sogenannte *superintuitionistische Logiken*. Es gibt in jeder superintuitionistischen Logik mehr Tautologien als in IPL. Syntaktisch entstehen diese Logiken aus IPL durch Hinzunahme von weiteren Axiomen als Abschluss unter uniformer Ersetzung und Modus Ponens. In diesem Sinne entsteht AL aus IPL durch Hinzunahme des starken Gesetz des ausgeschlossenen Dritten $(p \vee \neg p)$. Bei LC wird $(p \rightarrow q) \vee (q \rightarrow p)$ als Aximon hinzugenommen und bei KC das schwache Gesetz des ausgeschlossenen Dritten $(\neg\neg p \vee \neg p)$. Es gibt unendlich viele weitere superintuitionistischen Logiken, aber unsere Ergebnisse beschränken sich im Wesentlichen auf diese hier genannten.
Für $\mathrm{L} \in \{\mathrm{BPL}, \mathrm{FPL}, \mathrm{IPL}, \mathrm{KC}, \mathrm{LC}, \mathrm{AL}\}$ und $n \in \mathbb{N}$ bezeichnen wir mit $\mathrm{L}[n]$ die Logik L eingeschränkt auf VAR_n. Zum Beispiel ist $\mathrm{IPL}[1] = (\mathfrak{F}^{\mathsf{i}}[1], \mathfrak{K}^{\mathsf{i}}_{refl}[1])$, wobei $\mathfrak{K}^{\mathsf{i}}_{refl}[1] = \{(W, S, \xi, \{p\}) \mid (W, S, \xi, \{p\}) \in \mathfrak{K}^{\mathsf{i}}_{refl}\}$ ist.[7]

Einbettung in die Modallogik

Vergleicht man die Tabellen 2.1 und 2.2 mit den Übersichten über die Logiken dieser Arbeit, fällt schnell auf, dass es viele Parallelen zwischen den Modelleigenschaften einzelner Fragmente gibt. Haben die Rahmen einer modalen Modellklasse und einer intuitionistischen Modellklasse dieselben Eigenschaften, so gibt es in der modalen Modellklasse zusätzlich die Modelle, die keine monotone Belegungsfunktion haben. Dies legt eine mögliche Einbettung der intuitionistischen Logiken in die Modallogiken nahe. Die *Gödel-Tarski-Übersetzung* [Göd32] bettet intuitionistische Logiken in Modallogiken so ein, dass Gültigkeit erhalten bleibt. Gödel formulierte die Idee der Einbettung, bevor es die Kripke-Semantik gab. Zu diesem Zeitpunkt war der Zusammenhang zwischen intuitionistischen Logiken und ihren modalen Begleitern nicht so offensichtlich, wie er bei unserer Einführung auf Basis der Kripke-Semantik ist. Eine solche Einbettung ist Übersetzung 1 aus [Vis80].

[7] Diese Art der Bezeichnung verwenden wir für Modallogiken analog.

Definition 2.12 *Sei* $gt : \mathfrak{F}^{\mathsf{i}} \mapsto \mathfrak{F}$ *eine Funktion mit*

$$
\begin{array}{lll}
(1) & gt(\bot) & := \bot \ , \\
(2) & gt(p) & := p \wedge \Box p \ , \\
(3) & gt(\alpha \wedge \beta) & := gt(\alpha) \wedge gt(\beta) \ , \\
(4) & gt(\alpha \vee \beta) & := gt(\alpha) \vee gt(\beta) \ , \\
(5) & gt(\alpha \rightharpoonup \beta) & := \Box(gt(\alpha) \rightarrow gt(\beta)) \ .
\end{array}
$$

Dabei sind α *und* β *beliebige Formeln und* $p \in \mathrm{VAR}$.

Wir bezeichnen für eine intuitionistische Logik L^{i} die Modallogik L^{m} als *modalen Begleiter* von L^{i}, wenn für alle L^{i}-Formeln α gilt, dass $gt(\alpha)$ eine L^{m}-Formel ist. Zusätzlich muss gelten, dass α genau dann eine L^{i}-Tautologie ist, wenn $gt(\alpha)$ eine L^{m}-Tautologie ist. Visser [Vis80] zeigte, dass PrL in diesem Sinne ein modaler Begleiter von FPL ist. Die anderen Zusammenhänge aus Tabelle 2.3 können mit einfacher Induktion über den Formelaufbau gezeigt werden. Zur Einbettung von KC siehe auch [Boo93].

Intuitionistische Logik	Modaler Begleiter
BPL	K4
FPL	PrL
IPL	S4
KC	S4.2
LC	S4.3

Tabelle 2.3: Intuitionistische Logiken und ihre modalen Begleiter.

Es ist offensichtlich, dass ein modaler Begleiter nicht eindeutig bestimmt sein muss. In Tabelle 2.3 sind die üblichen Begleiter der Logiken aus dieser Arbeit gegenübergestellt.

2.1.3 Heyting-Semantik

Neben der Semantik basierend auf den Kripke-Modellen ist auch die Heyting-Semantik eine oft verwendete Semantik für intuitionistische Logiken, siehe dazu [Hey71]. Diese Semantik basiert auf Heyting-Algebren. Der Unterschied zu den Kripke-Modellen besteht darin, dass in einer Heyting-Algebra Formeln nicht mehr im ursprünglichen Sinne von Modellen erfüllt werden. Bei der Heyting-Semantik ist die Äquivalenz von Formeln der zentrale Punkt.
Für einige Ergebnisse in den Kapiteln 3 und 4 spielen Heyting-Algebren als Werkzeug eine wichtige Rolle. Wir geben deshalb eine kurze Einführung, die auf die für

uns wesentlichen Punkte beschränkt ist. Ausführliche Details können in [Joh82] nachgelesen werden.

Heyting-Algebren sind eine Verallgemeinerung der Booleschen Algebren. Boolesche Algebren wiederum sind spezielle Verbände. Ganz allgemein besteht ein Verband $(V, \sqcup, \sqcap)$ aus einer nicht leeren Menge V und zwei zweistelligen Operationen $\sqcup$ (*Vereinigung*) und $\sqcap$ (*Durchschnitt*). Dabei gelten für alle $x, y, z \in V$ folgende Kommutativitäts-, Assoziativitäts- und Absorptionsgesetze:

$$\begin{array}{llll} x \sqcup y = y \sqcup x & \text{und} & x \sqcap y = y \sqcap x & \text{Kommutativität,} \\ x \sqcup (y \sqcup z) = (x \sqcup y) \sqcup z & \text{und} & x \sqcap (y \sqcap z) = (x \sqcap y) \sqcap z & \text{Assoziativität,} \\ x \sqcup (x \sqcap y) = x & \text{und} & x \sqcap (x \sqcup y) = x & \text{Absorption.} \end{array}$$

Ein Verband $(V, \sqcup, \sqcap)$ induziert außerdem eine Halbordnung $\sqsubseteq$ über V. Es seien $x, y \in V$, dann gilt

$$x \sqsubseteq y \iff x \sqcap y = x \ .$$

Eine Heyting-Algebra $H = (V, \sqcup, \sqcap, \multimap, \bot)$ ist ein beschränkter Verband, bei dem es für zwei Elemente x und y immer ein größtes Element z in H gibt, so dass $x \sqcap z \sqsubseteq y$ gilt. Dieses Element z ist das *relative Pseudokomplement* von x bezüglich y und man schreibt $z = x \multimap y$. Das Element $\bot$ ist bezüglich der induzierten Halbordnung das kleinste Element. Mit $\top$ wird oft das größte Element bezeichnet.

Ein Beispiel lässt sich mit Hilfe von IPL $= (\mathfrak{F}^{\mathsf{i}}, \mathfrak{K}^{\mathsf{i}}_{refl})$ angeben. Für eine IPL-Formel α bezeichnet $[\alpha]$ die Menge aller $\mathfrak{F}^{\mathsf{i}}$-Formeln, die $\mathfrak{K}^{\mathsf{i}}_{refl}$-äquivalent zu α sind und [IPL] ist die Menge aller Äquivalenzklassen (bezüglich $\equiv_{\mathfrak{K}^{\mathsf{i}}_{refl}}$). Wir definieren die Operationen für $\alpha, \beta, \gamma \in \mathfrak{F}^{\mathsf{i}}$ in natürlicher Weise:

$$\begin{array}{lcl} [\alpha] \sqcup [\beta] & := & [\alpha \vee \beta] \ , \\ [\alpha] \sqcap [\beta] & := & [\alpha \wedge \beta] \ , \\ [\alpha] \multimap [\beta] & := & [\alpha \rightarrowtriangle \beta] \ . \end{array}$$

Die optische Ähnlichkeit der Operatoren verdeutlicht den engen Zusammenhang zwischen intuitionistischer Logik und Heyting-Algebra. Damit ist die Struktur $([\text{IPL}], \sqcup, \sqcap, \multimap, [\bot])$ eine Heyting-Algebra [Hey71]. Eine Formel $\alpha \in \mathfrak{F}^{\mathsf{i}}$ ist im Sinne dieser Heyting-Algebra gültig, wenn $[\alpha] = [\top]$ gilt. Aus der Konstruktion ist leicht zu sehen, dass die Menge der gültigen Formeln exakt die Menge der Tautologien in IPL ist. Es wird nochmal deutlich, dass es hier kein Analogon zur Erfüllt-sein-Relation der Kripke-Semantik gibt. Ein weiteres Beispiel ist der Rieger-Nishimura Verband. Dieser wird in Abschnitt 4.1 eingeführt, siehe dazu auch Abbildung 4.1.

2.2 Komplexitätstheorie

In der Komplexitätstheorie geht es im Wesentlichen um die Frage, wie viele und welche Ressourcen man benötigt, um ein gegebenes Problem zu lösen. Wir beschränken uns in dieser Arbeit auf Entscheidungsprobleme. Bei dieser Art von Problemen wird eine Eingabe entweder *akzeptiert* oder *abgelehnt*. In diesem Sinne fassen wir ein Problem als eine Teilmenge einer Grundmenge auf, zu der ein Element der Grundmenge – eine *Instanz* des Problems – dazugehören kann (eine *Ja-Instanz*) oder nicht (eine *Nein-Instanz*). Wir verwenden hauptsächlich die Terminologie der Turingmaschinen und die damit verbundenen Konzepte der Komplexitätsklassen und Reduktionen.
Im folgenden Abschnitt werden die für die Arbeit bedeutenden Komplexitätsklassen und Reduktionsbegriffe eingeführt. Im zweiten Teil stellen wir Entscheidungsprobleme vor, die für unsere Ergebnisse eine wichtige Rolle spielen.

2.2.1 Komplexitätsklassen und Reduktionen

Eine umfangreiche Einführung in die Grundlagen der Komplexitätstheorie, basierend auf der Turing-Berechenbarkeit, gibt es in [Pap94] und [Wec00]. Einige Komplexitätsklassen werden über Schaltkreise definiert, dazu gibt es eine Einführung in [Vol99]. Teilweise nutzen wir auch die Deskriptive Komplexität, die im Wesentlichen in [Imm99] erläutert wird. Wir geben einen kurzen Einblick in alle drei Varianten und konzentrieren uns dabei auf die für diese Arbeit wichtigen Details.

Komplexität auf Basis von Turingmaschinen

Bei den Ressourcen einer Turingmaschine betrachtet man in erster Linie die Laufzeit und den Speicherplatzbedarf. Zusätzlich spielt auch die Art der Turingmaschine eine Rolle – ist sie deterministisch, nichtdeterministisch oder alternierend.[8] Bei alternierenden Maschinen ist die Anzahl der Alternierungen eine weitere Ressource. Man betrachtet die Ressourcen meist in Abhängigkeit von der Eingabe. Im Folgenden geben wir die formale Definition der Begriffe an:
Es sei $f : \mathbb{N} \mapsto \mathbb{N}$ eine Funktion. Dann ist $\mathsf{DTIME}(f)$ (respektive $\mathsf{NTIME}(f)$) die Klasse aller Probleme, die von einer deterministischen (respektive nichtdeterministischen) Turingmaschine in der Laufzeit $\mathcal{O}(f(n))$ entschieden werden können. Hierbei bezeichnet n die Größe der Eingabe. $\mathsf{ATIME}(f)$ ist die Klasse aller Probleme, die von einer alternierenden Turingmaschine in Laufzeit $\mathcal{O}(f(n))$ entschieden werden können. Die Klassen $\mathsf{DSPACE}(f)$, $\mathsf{NSPACE}(f)$ und $\mathsf{ASPACE}(f)$ sind analog definiert, wobei die beschränkende Ressource nicht die Laufzeit, sondern der Speicherplatz ist. Für eine Funktion $g : \mathbb{N} \mapsto \mathbb{N}$ bezeichnet $\mathsf{ASPACE}(f)[g]$ die Teilklasse von $\mathsf{ASPACE}(f)$, in der nur Probleme enthalten sind, die mit höchstens

[8] Es gibt noch weitere Eigenschaften, die eine Turingmaschine haben kann, z.B. kann sie ein Orakel befragen oder probabilistisch sein. Diese Eigenschaften spielen in dieser Arbeit aber keine Rolle.

$\mathcal{O}(g(n))$ vielen Alternierungen entschieden werden können. Abkürzend verwenden wir auch $\mathsf{ALOGSPACE}[g]$ und meinen damit $\mathsf{ASPACE}(\log)[g]$. Statt einer Funktion f kann man in diesem Zusammenhang auch Funktionenklassen verwenden. Zum Beispiel ist $\mathsf{DTIME}(n^{\mathcal{O}(1)}) = \bigcup_{f \text{ ist ein Polynom in } n} \mathsf{DTIME}(f)$. Mit diesen Begriffen lassen sich folgende bekannte Komplexitätsklassen definieren:

$$
\begin{aligned}
\mathsf{L} &:= \mathsf{DSPACE}(\log(n)) \ , \\
\mathsf{NL} &:= \mathsf{NSPACE}(\log(n)) \ , \\
\mathsf{P} &:= \mathsf{DTIME}(n^{\mathcal{O}(1)}) \ , \\
\mathsf{NP} &:= \mathsf{NTIME}(n^{\mathcal{O}(1)}) \ , \\
\mathsf{PSPACE} &:= \mathsf{DSPACE}(n^{\mathcal{O}(1)}) \ .
\end{aligned}
$$

Jetzt können wir den ersten Reduktionsbegriff definieren. Seien G_1 und G_2 Grundmengen. Dann ist das Problem $P_1 \subseteq G_1$ in logarithmischem Platz zu $P_2 \subseteq G_2$ reduzierbar, wenn es eine Funktion $f : G_1 \mapsto G_2$ gibt, die in logarithmischem Platz berechnet werden kann[9] und für alle $x \in G_1$ gilt genau dann $x \in P_1$, wenn $f(x) \in P_2$. Abkürzend sagen wir P_1 ist *logspace-reduzierbar* zu P_2 und schreiben $P_1 \leq_{\mathrm{m}}^{\log} P_2$.[10]
Es sei $\mathsf{C} \in \{\mathsf{NL}, \mathsf{P}, \mathsf{NP}, \mathsf{PSPACE}\}$. Wir bezeichnen ein Problem P als C-*hart*, wenn $P' \leq_{\mathrm{m}}^{\log} P$ für alle Probleme P' aus C gilt. Das Problem P ist C-*vollständig*, wenn es C-hart und in C enthalten ist. Da die Logspace-Reduktion eine transitive Relation ist, zeigen wir C-Härte für ein Problem P in der Regel, indem wir eine Reduktion von einem C-harten Problem auf P angeben. Dieser Begriff der Härte überträgt sich auf alle weiteren Komplexitätsklassen, die NL enthalten.[11] Dass ein Problem P in einer Komplexitätsklasse enthalten ist, kann man auf zwei Arten zeigen. Der eine Weg besteht in der Angabe eines Algorithmus, der P entscheidet und der mit den Ressourcenbeschränkungen der Klasse auskommt. Der andere Weg ist wieder eine Reduktion. Man kann für ein in der Klasse enthaltenes Problem zeigen, dass P zu diesem reduzierbar ist. Wir bezeichnen C als *obere Schranke* für P, wenn P in C enthalten ist und als *untere Schranke*, wenn es C-hart ist.

Komplexität auf Basis von Schaltkreisen

Als nächstes betrachten wir Komplexitätsklassen, die über Schaltkreisen definiert sind. Eine ausführliche Einführung in dieses Thema bietet [Vol99]. Wir nutzen in dieser Arbeit hauptsächlich die Klassen NC^i und AC^i. Für $i \geq 0$ enthalten beide Klassen Probleme, die mit logspace-uniformen Schaltkreisfamilien polynomieller Größe und der Tiefe $\mathcal{O}((\log(n))^i)$ entschieden werden können. Eine Schaltkreis-

[9] Die Funktion f ist in logarithmischem Platz berechenbar, wenn für jedes $x \in G_1$ der Funktionswert $f(x)$ von einer Turingmaschine mit logarithmisch großem Arbeitsband berechnet werden kann.
[10] Das tiefgestellte m bei $\leq_{\mathrm{m}}^{\log}$ bedeutet, dass es sich um eine *many-one-Reduktion* handelt.
[11] L-Härte spielt in dieser Arbeit keine Rolle.

familie ist *logspace-uniform*, wenn es eine Turingmaschine gibt, die mit $\mathcal{O}(\log n)$ Speicherplatz überprüfen kann, ob ein Schaltkreis zur Familie gehört (n ist die Größe des Schaltkreises). Als Gatter sind Konjunktion, Disjunktion und Negation erlaubt. Der Unterschied zwischen den Klassen besteht darin, dass bei AC^i die Konjunktions- und die Disjunktionsgatter einen beliebigen Eingangsgrad haben, der auch von der Eingabegröße abhängen kann, während bei NC^i der Eingangsgrad auf zwei beschränkt ist. Das Negationsgatter hat grundsätzlich Eingangsgrad eins.
Formal werden NC^i und AC^i über die Schaltkreisklassen $\mathsf{SIZE\text{-}DEPTH}_{\mathfrak{B}}(s(n), d(n))$ definiert. Hierbei sind $s, d : \mathbb{N} \mapsto \mathbb{N}$ Funktionen und $\mathfrak{B}$ die Menge der erlaubten Gatter. $\mathsf{SIZE\text{-}DEPTH}_{\mathfrak{B}}(s(n), d(n))$ beschreibt dann die Klasse der Probleme, die von logspace-uniformen Schaltkreisfamilien mit der Größe $\mathcal{O}(s(n))$ und der Tiefe $\mathcal{O}(d(n))$ mit Gattern aus $\mathfrak{B}$ entschieden werden können. Die Größe ist die Gesamtzahl der Gatter und die Tiefe ist die Länge des längsten Pfades durch den Schaltkreis. Die Tiefe kann auch als Laufzeit aufgefasst werden, denn Gatter auf einem Pfad werden nacheinander durchlaufen. Seien $\mathfrak{B}_0 = \{\wedge, \vee, \neg\}$ und $\mathfrak{B}_1 = \{(\wedge^n)_{n\in\mathbb{N}}, (\vee^n)_{n\in\mathbb{N}}, \neg\}$[12], dann werden NC^i und AC^i für $i \geq 0$ wie folgt definiert:

$$\mathsf{NC}^i := \mathsf{SIZE\text{-}DEPTH}_{\mathfrak{B}_0}(n^{\mathcal{O}(1)}, (\log(n))^i) \ ,$$

$$\mathsf{AC}^i := \mathsf{SIZE\text{-}DEPTH}_{\mathfrak{B}_1}(n^{\mathcal{O}(1)}, (\log(n))^i) \ .$$

Schränkt man bei NC^1 die Uniformität stärker ein und fordert U_E^*-uniforme Schaltkreisfamilien[13], so lassen sich NC^1 und AC^1 auch mit dem Ressourcengebrauch einer Turingmaschine beschreiben. Bei AC^1 fallen logspace- und U_E^*-Uniformität zusammen [Vol99].

Theorem 2.13 *Es gilt*

(1) $\mathsf{NC}^1 = \mathsf{ATIME}(\log(n))$ [*Ruz81*],

(2) $\mathsf{AC}^1 = \mathsf{ASPACE}(\log(n))[\log(n)]$ [*Coo85*].

Ob diese Gleichheit auch für logspace-uniformes NC^1 gilt, ist ein offenes Problem. Im weiteren Verlauf der Arbeit gehen wir bei Verwendung von NC^1 grundsätzlich von der U_E^*-uniformen Variante aus. An Hand dieser Charakterisierung ist auch leicht verständlich, warum Probleme, die in NC^1 lösbar sind, als *effizient parallelisierbar* gelten. Intuitiv gesprochen gibt die Tiefe der Schaltkreise die Laufzeit an und die Alternierungen werden durch Parallelisierung simuliert. Deswegen lassen sich Probleme in NC^1 mit parallelen Turingmaschinen echt schneller lösen als mit

[12] $\wedge^n$ und $\vee^n$ bezeichnen die n-stelligen Varianten von $\wedge$ und $\vee$.

[13] Eine Schaltkreisfamilie $\mathcal{C} = (C)_{n\in\mathbb{N}}$ ist U_E^*-uniform, wenn ihre zugehörige *erweiterte Verbindungssprache* $L_{EC}(\mathcal{C})$ von einer alternierenden Turingmaschine in $\mathcal{O}(\log s(n))$ Platz und $\mathcal{O}(d(n))$ Zeit akzeptiert wird. Die genaue Definition ist in [Vol99, Kapitel 2.6, Definitionen 2.42 bis 2.44] zu finden.

nicht parallelen (deterministischen) Turingmaschinen, die dafür polynomielle Zeit benötigen.[14] Mehr dazu findet man unter anderem in [Coo85, KR90].
Mit Hilfe von Schaltkreisen lässt sich auch ein weiterer Reduktionsbegriff definieren. Ein Problem P_1 ist *konstante-Tiefe-reduzierbar* oder *cd-reduzierbar* zu einem Problem P_2, wenn es eine logtime-uniforme Schaltkreisfamilie[15] gibt, die P_2 entscheidet und folgende Eigenschaft hat: Die Schaltkreise haben konstante Tiefe und polynomielle Größe. Neben den normalen Gattern dürfen zusätzlich noch P_1-Gatter vorkommen. Ein P_1-Gatter ist hierbei ein Gatter, das eine P_1-Instanz als Eingabe bekommt und genau dann 1 ausgibt, wenn es sich um eine Ja-Instanz handelt. Wir schreiben in diesem Fall $P_1 \leq^{\text{cd}} P_2$. Offensichtlich ist dieser Reduktionsbegriff strenger als der der logspace-Reduktion. Um AC^1-Härte zu zeigen, reicht es, die logspace-Reduktion zu verwenden. Will man für NC^1 einen Härtebeweis mittels Reduktion führen, muss man eine schwächere Reduktion verwenden. Anderenfalls könnte das Problem bereits in der Reduktionsfunktion gelöst werden (bzw. mit deren Ressourcen). In diesem Fall verwenden wir die *cd*-Reduktion.
Eine weitere Schaltkreisklasse ist TC^0. Sie spielt in dieser Arbeit eine untergeordnete Rolle. TC^0-Schaltkreise haben wie AC^0-Schaltkreise konstante Tiefe und polynomielle Größe. Die Gatter haben unbeschränkten Eingangsgrad und es gibt zusätzliche *Threshold-Gatter*[16], siehe dazu [Vol99].

Weitere Komplexitätsklassen und Zusammenfassung

Neben den schon eingeführten Klassen verwenden wir außerdem noch die Klassen LOGdetCFL und LOGCFL. Diese können mit Hilfe von kontextfreien Sprachen definiert werden. Die Klasse LOGdetCFL (respektive LOGCFL) ist die Klasse aller Probleme, die zum Wortproblem einer deterministischen (respektive nichtdeterministischen) kontextfreien Sprache logspace-reduzierbar sind. Auch diese Klassen können über Ressourcengebrauch einer Turingmaschine charakterisiert werden.

Theorem 2.14 ([Coo71a]) LOGdetCFL *(respektive* LOGCFL*) ist die Klasse aller Probleme, die von einer deterministischen (respektive nichtdeterministischen) Turingmaschine in polynomieller Zeit, mit logarithmischem Speicherplatz und einem zusätzlichen Keller entschieden werden können.*

Eine weitere Art, sich der Komplexitätstheorie zu nähern, ist die Deskriptive Komplexität. Begründet wurde diese mit einer Beschreibung von NP auf Basis von Prädikatenlogik zweiter Ordnung durch Fagin 1974 [Fag74]. Weiter befasste sich damit unter anderem Immerman in [Imm79, Imm89]. Eine umfangreiche Einführung ist in [Imm99] zu finden. Die Klasse FO ist die kleinste Klasse der

[14] Nach derzeitigem Kenntnisstand gilt $\mathsf{NC}^1 \subsetneq \mathsf{P}$ als sehr wahrscheinlich, ist aber noch nicht bewiesen.

[15] Logtime-Uniformität bedeutet, dass es eine Turingmaschine gibt, die in $\mathcal{O}(\log n)$ Zeit überprüfen kann, ob ein Schaltkreis zur Familie gehört, wobei n die Größe des Schaltkreises ist.

[16] Ein Threshold-Gatter gibt genau dann `true` aus, wenn nicht mehr Eingänge auf `false` geschaltet sind als auf `true`.

Deskriptiven Komplexität. Für unsere Verwendung von FO genügt folgende Charakterisierung: FO ist die Klasse aller Probleme, die durch eine prädikatenlogische Formel erster Stufe entschieden werden können. Das bedeutet, es gibt eine Formel, die das Problem charakterisiert. Instanzen für das Problem werden in Strukturen kodiert, über denen man die Formel auswerten kann. Die Ja-Instanzen werden genau durch die Strukturen dargestellt, unter denen die Formel wahr ist. Unter den Strukturen der Nein-Instanzen ist die Formel falsch. Immerman zeigte für FO den folgenden Zusammenhang.

Theorem 2.15 ([Imm99, Theorem 5.22]) *Es gilt* $\mathsf{FO} = \mathsf{AC}^0$.

Dieses Theorem sagt aus, dass es für jedes Problem in AC^0 eine prädikatenlogische Formel gibt, durch die es charakterisiert wird.
Für die in diesem Abschnitt eingeführten Komplexitätsklassen gilt die folgende Inklusionsstruktur.

Theorem 2.16 *Es gilt*

$$\mathsf{AC}^0 \subset \mathsf{TC}^0 \subseteq \mathsf{NC}^1 \subseteq \mathsf{L} \begin{array}{c} \subsetneqq \\ \subsetneqq \end{array} \begin{array}{c} \mathsf{LOGdetCFL} \\ \mathsf{NL} \end{array} \begin{array}{c} \subsetneqq \\ \subsetneqq \end{array} \mathsf{LOGCFL} \subseteq \mathsf{AC}^1 \subseteq \mathsf{P} \begin{array}{c} \subsetneqq \\ \subsetneqq \end{array} \begin{array}{c} \mathsf{NP} \\ \mathsf{coNP} \end{array} \begin{array}{c} \subsetneqq \\ \subsetneqq \end{array} \mathsf{PSPACE}\,.$$

Die echte Inklusion $\mathsf{AC}^0 \subset \mathsf{TC}^0$ folgt aus [Smo87] und $\mathsf{LOGCFL} \subseteq \mathsf{AC}^1$ folgt aus [Ruz80]. Alle anderen Inklusionen folgen direkt aus der Definition der Komplexitätsklassen oder lassen sich leicht zeigen.

2.2.2 Vollständige Probleme

Für unsere Komplexitätsresultate müssen wir häufig Reduktionen angeben. Dafür benötigen wir Probleme, auf die wir oder von denen wir reduzieren können. Solche Probleme, deren Komplexität bereits bekannt ist, oder die nicht aus dem Gebiet der Logik kommen, wollen wir in diesem Abschnitt vorstellen. Im ersten Teil betrachten wir alternierende Graphen und Wege. Um den längsten Pfad geht es im zweiten Teil. Mit der Formelauswertung in der Aussagenlogik (und damit verwandten Problemen) beschäftigen wir uns im letzten Teil.

Alternierende Wegsuche

Pfade in Graphen sind eine lineare Aneinanderreihung von verbundenen Knoten, die besucht werden. Basis bei der alternierenden Wegsuche ist ein (alternierender) Graph, dessen Knotenmenge sich in Existenz- und Universalknoten unterteilt. Bei einem alternierenden Pfad geht von jedem Existenzknoten auf diesem Pfad eine Kante aus, über die man alternierend zum Ziel kommt. Von jedem Universalknoten auf dem Pfad kommt man über jede ausgehende Kante alternierend zum Ziel. So wie ein normaler Pfad einen Weg durch den Konfigurationsgraphen einer nichtdeterministischen Turingmaschine zu einem akzeptierenden Endzustand beschreiben

kann, kann man mit alternierenden Pfaden durch die Konfigurationsgraphen von alternierenden Turingmaschinen navigieren.
Für das Alternierende-Graph-Erreichbarkeits-Problem (AGEP) zeigten Chandra, Kozen und Stockmeyer 1981 [CKS81], dass es P-vollständig ist (siehe hierzu auch [Imm82]). Später zeigte Immerman die P-Vollständigkeit auch via *cd*-Reduktionen [Imm86]. Dieses Problem bleibt P-vollständig, wenn man sich auf bipartite Graphen[17] beschränkt [GHR95]. Wir arbeiten ausschließlich mit einer bipartiten Variante von AGEP und verwenden Graphem, bei denen sich Schichten mit Existenz- und Universalknoten immer abwechseln.

Definition 2.17 *Einen gerichteten bipartiten Graphen $G = (V, E)$ mit den Partitionen $V_\exists$ und $V_\forall$ ($V = V_\exists \cup V_\forall \neq \varnothing$, $V_\exists \cap V_\forall = \varnothing$ und $E \cap (V_\exists^2 \cup V_\forall^2) = \varnothing$) bezeichnen wir als alternierenden Graphen. Die Knoten aus $V_\exists$ heißen $\exists$-Knoten (Existenzknoten), die aus $V_\forall$ sind die $\forall$-Knoten (Universalknoten).*

In alternierenden Graphen gibt es *alternierende Pfade*. Deren Existenz soll durch die Eigenschaft $aPfad_G(s,t)$ ausgedrückt werden. Sie gibt an, dass es in G einen alternierenden Pfad vom Knoten s zum Knoten t gibt.

Definition 2.18 *Es seien $G = (V, E)$ ein alternierender Graph und $s, t \in V$ Knoten.*

(1) *Wenn $s = t$ ist, dann gilt $aPfad_G(s,t)$.*

(2) *Ist $s \in V_\exists$, so gilt:* $aPfad_G(s,t) \iff \exists u \in V_\forall, (s,u) \in E : aPfad_G(u,t)$.

(3) *Ist $s \in V_\forall$, so gilt:* $aPfad_G(s,t) \iff \forall u \in V_\exists, (s,u) \in E : aPfad_G(u,t)$.

Das Problem AGEP ist die Frage, ob es in einem alternierenden Graphen einen alternierenden Pfad von einem Start- zu einem Zielknoten gibt.

Problem:	AGEP
Eingabe:	$\langle G, s, t\rangle$, wobei $G = (V_\exists \cup V_\forall, E)$ ein alternierender Graph ist und $s, t \in V$ Knoten sind.
Frage:	Gilt $aPfad_G(s,t)$?

Theorem 2.19 ([CKS81, GHR95]) *Das Problem* AGEP *ist* P*-vollständig.*

Der Beweis für die untere Schranke basiert im Wesentlichen auf der Tatsache, dass $\mathsf{P} = \mathsf{ASPACE}(\log(n))[n^{\mathcal{O}(1)}]$ [CKS81] gilt. Ein alternierender Graph kann genutzt werden, um den Konfigurationsgraphen einer alternierenden Turingmaschine zu simulieren. Dann drückt die *aPfad*-Eigenschaft aus, dass es einen akzeptierenden

[17] Ein Graph ist bipartit, wenn sich seine Knotenmenge so in zwei disjunkte Partitionen zerlegen lässt, dass keine zwei Knoten innerhalb einer Partition durch eine Kante verbunden sind.

Pfad in diesem Konfigurationsgraphen gibt. Die obere Schranke folgt ebenfalls direkt aus $\mathsf{P} = \mathsf{ASPACE}(\log(n))[n^{\mathcal{O}(1)}]$.
Wir verwenden eine eingeschränkte Version von AGEP und fordern zusätzlich, dass es sich bei den Graphen um sogenannte *Schichtgraphen* handelt. Solche Graphen sind bipartit und es wechseln sich immer Schichten mit ∀-Knoten und mit ∃-Knoten ab.

Definition 2.20 *Ein alternierender Graph $G = (V, E)$ mit der Partitionierung $V = V_\exists \cup V_\forall$ ist ein alternierender Schichtgraph mit $m > 0$ Schichten, wenn zusätzlich $V = V_1 \cup V_2 \cup \cdots \cup V_m$ gilt, wobei*

(1) $V_i \cap V_j = \varnothing$, *wenn* $i \neq j$,

(2) $V_\exists = \bigcup_{i \leq m,\, i \text{ ungerade}} V_i$,

(3) $V_\forall = \bigcup_{i \leq m,\, i \text{ gerade}} V_i$,

(4) $E \subseteq \bigcup_{i=1}^{m-1} V_i \times V_{i+1}$,

(5) $\forall i = 1, 2, \ldots, m-1\ \forall v \in V_i\ \exists v' \in V_{i+1} : (v, v') \in E$.

Eigenschaft (4) drückt aus, dass alle Kanten nur zwischen einer Schicht und der direkt nachfolgenden Schicht verlaufen. Eigenschaft (5) besagt, dass alle Knoten, die nicht aus der letzten Schicht (V_m) sind, einen positiven Ausgangsgrad haben. Für diese Graphen definieren wir ebenfalls ein alternierendes Erreichbarkeitsproblem.

Problem:	ASGEP
Eingabe:	$\langle G, s, t\rangle$, wobei $G = (V, E)$ ein alternierender Schichtgraph mit $m > 0$ Schichten ist und $s \in V_1$ und $t \in V_m$ Knoten sind.
Frage:	Gilt $aPfad_G(s, t)$?

Ein Beispiel eines Schichtgraphen und eines alternierenden Pfades ist in Abbildung 2.2 zu sehen.

Theorem 2.21 *Das Problem* ASGEP *ist* P*-vollständig.*

Beweis. Da Schichtgraphen spezielle alternierende Graphen sind, ist ASGEP ebenfalls in P enthalten.
Für die P-Härte zeigen wir AGEP $\leq_{\mathrm{m}}^{\log}$ ASGEP. Sei $\langle G, s, t\rangle$ ein Instanz von AGEP mit $G = (V, E)$ und $V = V_\exists \cup V_\forall$. Ohne Einschränkungen gehen wir davon aus, dass der Startknoten s ein Existenzknoten ist ($s \in V_\exists$) und alle Universalknoten einen positiven Ausgangsgrad haben. Wir konstruieren einen alternierenden Schichtgraphen $G^{SG} = (V^{SG}, E^{SG})$ mit $m = 2 \cdot \lceil \frac{|V|}{2} \rceil$ Schichten wie folgt:

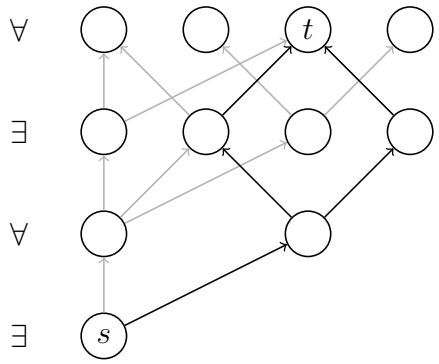

Abbildung 2.2: Ein Beispiel für einen alternierenden Schichtgraphen und einen alternierenden Pfad von s nach t. Die Kanten des Pfades sind schwarz gezeichnet, die anderen grau.

$$V_i := \begin{cases} \{\langle v,i\rangle \mid v \in V_\exists\} \cup \{\langle d,i\rangle\} & \text{für } 1 \leq i < m \text{ und } i \text{ ungerade}\ , \\ \{\langle v,i\rangle \mid v \in V_\forall\} \cup \{\langle d,i\rangle\} & \text{für } 1 < i \leq m \text{ und } i \text{ gerade}\ . \end{cases}$$

Dabei gilt $d \notin V$. Wir brauchen die Knoten $\langle d,i\rangle$, um für Existenzknoten aus V, die keine Nachfolger haben, einen positiven Ausgangsgrad im Schichtgraphen zu garantieren. Sie fungieren dabei als eine Art Dummyknoten, von denen man nicht zum Zielknoten $\langle t,m\rangle$ kommen kann. Wir erzeugen von den Partitionen $V_\exists$ und $V_\forall$ viele Kopien, die später durch Kanten immer abwechselnd miteinander verbunden werden:

$$V_\exists^{SG} := \bigcup_{i=1,\, i \text{ ungerade}}^{m-1} V_i\ ,$$

$$V_\forall^{SG} := \bigcup_{i=2,\, i \text{ gerade}}^{m} V_i\ .$$

Die Kanten in E^{SG} setzen sich aus drei Teilen zusammen:

$$E_i := \left\{\left(\langle v,i\rangle, \langle u,i+1\rangle\right) \;\middle|\; v \neq t \text{ und } (v,u) \in E\right\} \text{ für } 1 \leq i < m\ ,$$

$$E^t := \left\{\left(\langle t,i\rangle, \langle t,i+1\rangle\right) \;\middle|\; 1 \leq i < m\right\}\ ,$$

$$E^d := \left\{\left(\langle v,i\rangle, \langle d,i+1\rangle\right) \;\middle|\; i < m, i \text{ ungerade}, v \neq t, \forall u \in V : (v,u) \notin E\right\} \cup \left\{\left(\langle d,i\rangle, \langle d,i+1\rangle\right) \;\middle|\; 1 \leq i < m\right\}\ .$$

Dabei enthalten die E_i die Kanten, die gemäß der Kanten des Ausgangsgraphen G die Schichten miteinander verbinden. E^t sorgt dafür, dass ein Pfad, der einmal beim Zielknoten ankommt, diesen nicht wieder verlässt. E^d ist ein technisches Hilfsmittel, um zu garantieren, dass alle Knoten (außer die der letzten Schicht) einen positiven Ausgangsgrad haben. Die Bestandteile des Graphen G^{SG} sind da-

mit definiert:

$$\begin{aligned} V^{SG} &:= V_{\exists}^{SG} \cup V_{\forall}^{SG} \ , \\ E^{SG} &:= E^t \cup E^d \cup E_1 \cup E_2 \cup \cdots \cup E_{m-1} \ . \end{aligned}$$

Der Graph G^{SG} ist offensichtlich ein Schichtgraph nach Definition 2.20. Wir zeigen jetzt, dass es von $\langle s, 1\rangle$ nach $\langle t, m\rangle$ genau dann einen alternieren Pfad gibt, wenn in G von s nach t ein solcher Pfad existiert.

Behauptung 2.1 *Es gilt $aPfad_G(s,t)$ genau dann, wenn $aPfad_{G^{SG}}(\langle s,1\rangle, \langle t,m\rangle)$.*

Beweis der Behauptung. Der Kern der Idee besteht darin, dass man in G^{SG} von Schicht zu Schicht geht, statt zwischen den Partitionen $V_\exists$ und $V_\forall$ hin und her zu springen. Kommt man dabei in einer Kopie des Zielknotens t an, muss man nur noch durch alle Schichten bis in die oberste (Schicht m) gehen (dafür E^t). Den eigentlichen Beweis der Behauptung führen wir mit vollständiger Induktion über die Anzahl der Alternierungen auf einem Pfad.
Aus der Konstruktion folgt sofort, dass von jeder Kopie $\langle t, i\rangle$, $1 \leq i \leq m$, von t ein alternierender Pfad zu $\langle t, m\rangle$ existiert. Zusammen mit

$$aPfad_G(s,t) \text{ mit } a \text{ Alternierungen} \iff aPfad_{G^{SG}}(\langle s,1\rangle, \langle t, a+1\rangle)$$

für $0 \leq a < m$ folgt die Behauptung. Dafür zeigen wir jetzt für $x \in V$ und $b \leq a$

$$aPfad_G(x,t) \text{ mit } b \text{ Alternierungen} \iff aPfad_{G^{SG}}(\langle x, a+1-b\rangle, \langle t, a+1\rangle)$$

mit Induktion über b.
Der Induktionsanfang ist trivial, da aus $b = 0$ sofort $x = t$ folgt.
Im Induktionsschritt unterscheiden wir die Fälle $x \in V_\exists$ und $x \in V_\forall$. Für $x \in V_\exists$ gelten folgende Äquivalenzen:

$$\begin{aligned} & aPfad_G(x,t) \text{ mit } b \text{ Alternierungen} && \text{(i)} \\ \Leftrightarrow\ & \exists y \in V_\forall, (x,y) \in E : aPfad_G(y,t) \text{ mit } b-1 \text{ Alternierungen} && \text{(ii)} \\ \Leftrightarrow\ & \exists y \in V_\forall, (x,y) \in E : aPfad_{G^{SG}}(\langle y, a+1-(b-1)\rangle, \langle t, a+1\rangle) && \text{(iii)} \\ \Leftrightarrow\ & \exists y \in V_\forall, (\langle x, a+1-b\rangle, \langle y, a+1-(b-1)\rangle) \in E^{SG} : \\ & \qquad aPfad_{G^{SG}}(\langle y, a+1-(b-1)\rangle, \langle t, a+1\rangle) && \text{(iv)} \\ \Leftrightarrow\ & \exists z \in V_\forall^{SG} \setminus \{\langle d, \cdot\rangle\}, (\langle x, a+1-b\rangle, z) \in E^{SG} : \\ & \qquad aPfad_{G^{SG}}(z, \langle t, a+1\rangle) && \text{(v)} \\ \Leftrightarrow\ & aPfad_{G^{SG}}(\langle x, a+1-b\rangle, \langle t, a+1\rangle) && \text{(vi)} \end{aligned}$$

Die Äquivalenz zwischen (i) und (ii) folgt aus Definition 2.18 der alternierenden Pfade, die zwischen (ii) und (iii) aus der Induktionsvoraussetzung. Die Konstruktion von G^{SG} liefert die Äquivalenzen zwischen (iii) und (iv) und (iv) und (v).

Dass z nur aus Schicht $a + 1 - (b - 1)$ kommen kann, liegt insbesondere daran, dass die Kanten in G^{SG} ausschließlich zwischen direkt benachbarten Schichten verlaufen. Dass z kein Dummyknoten $\langle d, \cdot \rangle$ sein kann, folgt aus $(x, y) \in E$. Die letzte Äquivalenz folgt wieder aus Definition 2.18.
Der Beweis im Fall $x \in V_\forall$ verläuft analog und es folgt Behauptung 2.1. ▪
Der Graph G^{SG} kann aus G unter Benutzung von logarithmischem Platz konstruiert werden. Mit Hilfe von Behauptung 2.1 folgt direkt AGEP $\leq_m^{\log}$ ASGEP und mit Theorem 2.19 gilt, dass ASGEP P-vollständig ist. □

Eine einfache Anwendung dieses Theorems ist der Beweis von Theorem 2.32. Wir zeigen dort, wie man mit der Auswertung einer Formel einen alternierenden Pfad durch ein Kripke-Modell zeichnet. Das Problem ASGEP wird hauptsächlich in Kapitel 5 verwendet, um die P-härte von verschiedenen Formelauswertungsproblemen zu beweisen. Im Folgenden definieren wir eine AC^1-vollständige Variante von ASGEP.

Problem:	$\text{ASGEP}_{\log}$
Eingabe:	$\langle G, s, t \rangle$, wobei $G = (V, E)$ ein alternierender Schichtgraph mit $0 < m \leq c \cdot \log(\lvert V \rvert)$ Schichten[18] ist und $s \in V_1$ und $t \in V_m$ Knoten sind.
Frage:	Gilt $aPfad_G(s, t)$?

Im Unterschied zu ASGEP haben die Instanzen von $\text{ASGEP}_{\log}$ nur logarithmisch viele Schichten in der Anzahl der Knoten.

Theorem 2.22 $\text{ASGEP}_{\log}$ *ist* AC^1*-vollständig.*

Beweis. Die obere Schranke kann im Wesentlichen genauso gezeigt werden, wie P als obere Schranke von AGEP. Für eine Instanz $\langle G = (V, E), s, t \rangle$ ist die Länge des alternierenden Pfades maximal logarithmisch in der Anzahl der Knoten. Aus der Definition der alternierenden Pfade (Definition 2.18) kann man direkt einen Algorithmus konstruieren, dem die Ressourcen von $\mathsf{ASPACE}(\log(n))[\log(n)]$ genügen und der $\text{ASGEP}_{\log}$ entscheidet. Dabei ist n die Eingabegröße. Es folgt $\text{ASGEP}_{\log} \in \mathsf{AC}^1$ direkt, da $\mathsf{AC}^1 = \mathsf{ASPACE}(\log(n))[\log(n)]$ [Coo85] gilt.
AC^1 als untere Schranke von $\text{ASGEP}_{\log}$ zeigen wir mit folgender Behauptung.

Behauptung 2.2 *Für jedes Problem A in AC^1 gibt es eine Funktion f, die Instanzen x von A auf Instanzen $f(x) = \langle G_x, s_x, t_x \rangle$ von $\text{ASGEP}_{\log}$ abbildet. Dabei ist f in logarithmischem Platz berechenbar und es gilt für alle Instanzen x von A, dass $x \in A$ genau dann gilt, wenn $f(x) \in \text{ASGEP}_{\log}$ ist.*

[18] Die Konstante c ist unabhängig von der Eingabe.

Beweis der Behauptung. Es seien A eine AC^1-entscheidbare Menge und $(C_n)_{n\in\mathbb{N}}$ die zugehörige Schaltkreisfamilie. Die Funktion f übersetzt den Schaltkreis $C_{|x|}$ und die Eingabe x direkt in einen alternierenden Schichtgraphen G_x wie folgt:
Ohne Einschränkungen kann man davon ausgehen, dass sich in $C_{|x|}$ die $\wedge$- und die $\vee$-Gatter abwechseln und dass es keine Negationsgatter gibt. Die Schichten aus $\vee$-Gattern sind die $\exists$-Schichten und die aus $\wedge$-Gattern sind die $\forall$-Schichten. Der Startknoten s_x ist das Ausgabegatter von $C_{|x|}$ und der Zielknoten t_x repräsentiert genau die Eingabebits, die mit 1 belegt sind. Um aus den Kanten im Konfigurationsgraphen die Kanten des Modells zu machen, wird ihre Richtung einfach umgekehrt. Da $(C_n)_{n\in\mathbb{N}}$ logspace-uniform ist und $C_{|x|}$ logarithmische Tiefe hat, ist f in logarithmischem Platz berechenbar und $\langle G_x, s_x, t_x\rangle$ ist eine Instanz von $\textsc{Asgep}_{\log}$. Nach Konstruktion ist es offensichtlich, dass es in G_x einen alternierenden Pfad von s_x nach t_x genau dann gibt, wenn $C_{|x|}$ die Eingabe x akzeptiert.
■
Mit Behauptung 2.2 wurde gezeigt, dass jedes Problem in AC^1 zu $\textsc{Asgep}_{\log}$ reduzierbar ist. Es folgt, dass $\textsc{Asgep}_{\log}$ AC^1-hart ist. □

Die logarithmische Variante $\textsc{Asgep}_{\log}$ von Asgep bildet die Basis unseres Beweises der AC^1-Härte von IPL[1]-FA (Abschnitt 4.3, Theorem 4.22). Aus $\mathsf{P} = \mathsf{ASPACE}(\log(n))[n^{\mathcal{O}(1)}]$ und $\mathsf{AC}^1 = \mathsf{ASPACE}(\log(n))[\log(n)]$ kann man auch ableiten, dass die Idee, $\textsc{Asgep}_{\log}$ für die AC^1-Härte zu verwenden, ganz analog zur Verwendung von Asgep für die P-Härte ist.
Wir reduzieren in dieser Arbeit oftmals auf das Problem Asgep bzw. $\textsc{Asgep}_{\log}$ und verwenden dabei eine logspace-Reduktion. Häufig wird im Verlauf solch einer Reduktion ein transitiver Graph benötigt. Die transitive Hülle eines Graphen lässt sich aber nicht in logarithmischem Platz berechnen. Deswegen verwenden wir eine Verallgemeinerung – die *pseudotransitive Hülle*.

Definition 2.23 *Es sei $G = (V, E)$ mit m Schichten $V = V_1 \cup V_2 \cup \cdots \cup V_m$ ein Schichtgraph. Sei weiter $V_{\geq i} = \bigcup_{j=i,i+1,\ldots,m} V_j$. Die pseudotransitive Hülle von G ist dann der Graph $G' = (V, E')$ wobei*

$$E' \quad := \quad E \quad \cup \quad \bigcup_{i=1,2,\ldots,m-2} (V_i \times V_{\geq i+2}) \ .$$

Die pseudotransitive Hülle lässt sich in logarithmischem Platz berechnen.

Der längste Pfad

Ein weiteres Graphen-Problem ist die Frage nach dem längsten Pfad in einem Graphen. Der Begriff *Pfad* bezieht sich hier auf die nicht alternierende Variante.

Definition 2.24 *Es seien $G = (V, E)$ ein Graph und $s, t \in V$ Knoten in diesem Graph. Ein Pfad von s nach t existiert, wenn*

(1) *$s = t$ gilt, dieser Pfad hat dann die Länge*[19] *1, oder*

(2) *es einen Knoten $x \in V$ gibt, so dass zwischen s und x ein Pfad existiert und (x, t) eine Kante in E ist. Dieser Pfad hat dann die Länge des Pfades zwischen s und x plus 1.*

Das Problem des längsten Pfades ist wie folgt definiert.

Problem:	LPfad
Eingabe:	$\langle G, s, n\rangle$, wobei $G = (V, E)$ ein gerichteter kreisfreier Graph ist, $s \in V$ ein Knoten und $n \in \mathbb{N}$ ist.
Frage:	Hat der längste Pfad in G, der in s startet, die Länge n?

Für LPfad ist folgendes Komplexitätsresultat bekannt.

Theorem 2.25 ([JT07]) *Das Problem* LPfad *ist* NL*-vollständig.*

Wir verwenden dieses Problem in den Beweisen der Theoreme 5.8 und 5.9.

Formelauswertung in der Aussagenlogik

Die Auswertung aussagenlogischer Formeln ohne Variablen ist das Boolean Formula Value Problem.

Problem:	BFVP$[O]$
Eingabe:	φ ist eine aussagenlogische Formel ohne Variablen, die nur Operatoren aus O enthält.
Frage:	Ist φ gültig?

In der Literatur wurde dieses Problem bereits umfangreich untersucht. Eine Verallgemeinerung von BFVP$[\wedge, \vee, \neg]$ ist das Wortproblem für kontextfreie Klammersprachen. Lynch [Lyn77] zeigte, dass dieses Wortproblem und damit auch BFVP$[\wedge, \vee, \neg]$ in L ist. Buss [Bus87, BCGR90] verbesserte das Resultat und zeigte, dass das Wortproblem für jede kontextfreie Sprache in NC^1 entschieden werden kann. Für BFVP$[\wedge, \vee, \neg]$ konnte er auch die NC^1-Härte zeigen. Für kontextfreie Klammersprachen hängt die untere Schranke von der konkreten Wahl der Sprache ab. Zum Beispiel kann man die Menge aller syntaktisch korrekten Formeln der Aussagenlogik durch eine kontextfreie Klammersprache beschreiben. Für diese Sprache ist das Wortproblem TC^0-vollständig, da man dabei im Wesentlichen die Korrektheit der Klammerstruktur mittels Zählens überprüfen muss (siehe auch Lemma 2.29). Mehr dazu in [Vol99].

[19] Wir zählen bei der Pfadlänge die besuchten Knoten, in der Literatur werden oft auch die Kanten gezählt – die Länge wäre dann um eins kürzer.

Theorem 2.26 ([Bus87]) *Das Problem* $\mathrm{BFVP}[\wedge, \vee, \neg]$ *ist* NC^1*-vollständig.*

Da die Formeln sowohl $\wedge$ als auch $\vee$, aber keine Variablen enthalten dürfen, folgt dieses Resultat auch für monotone Formeln[20] [BM95].
Will man eine Formel mit Variablen unter einer Belegung auswerten, ersetzt man jede Variable durch eine Konstante entsprechend der Belegung und erhält so eine Formel ohne Variablen. Für monotone Formeln mit Variablen ergibt sich damit auch, dass das Erfüllbarkeitsproblem NC^1-vollständig ist [Sch07b]. Es muss dazu nur die Belegung überprüft werden, die allen Variablen `true` zuordnet.
Eine weitere Verfeinerung wurde auf Basis des Postschen Verbandes [Pos41] von Schnoor [Sch10] vorgenommen. Da eine Reihe seiner Ergebnisse unter NC^1 liegen, geht er von sogenannten *Promise-Problemen* aus. Bei diesen Problemen muss die Eingabe nicht mehr auf syntaktische Korrektheit geprüft werden. Diese Prüfung erfordert die Ressourcen von TC^0 (siehe auch Lemma 2.29). In einigen Fällen würde der Aufwand der Überprüfung den der eigentlichen Lösung des Problems dominieren. Zuerst verallgemeinert er von zweistelligen auf Operatoren mit beliebig vielen Stellen und zeigt, dass unabhängig von der Wahl der Operatoren das Auswertungsproblem immer in NC^1 ist. Für unsere Arbeit ist folgendes Resultat relevant [Sch10, Theorem 12].

Theorem 2.27 *Das Problem* $\mathrm{BFVP}[\bot, \to]$ *ist* NC^1*-vollständig.*

Wir interessieren uns in dieser Arbeit hauptsächlich für Probleme, deren Komplexität NC^1 nicht unterschreitet. Deswegen gehen wir auch grundsätzlich nicht von Promise-Problemen aus. Die syntaktische Überprüfung der Eingabe bei unseren Problemen ist im Vergleich zur eigentlichen Lösung sehr einfach (sieht dazu Lemmata 2.28 und 2.29).

2.3 Entscheidungsprobleme in der Logik

Das Erfüllbarkeitsproblem und das Gültigkeits- oder Tautologieproblem sind die am meisten untersuchten Entscheidungsprobleme in der Logik. Das bekannteste Resultat hierzu ist die NP-Vollständigkeit des Erfüllbarkeitsproblems der Aussagenlogik [Coo71b]. In dieser Arbeit geht es im Wesentlichen um das Formelauswertungsproblem intuitionistischer Logiken. Dabei geben wir aber auch eine Reihe von Nebenresultaten an, die sich auf andere Probleme beziehen. In Abschnitt 2.3.1 definieren wir die relevanten Entscheidungsprobleme und in Abschnitt 2.3.2 geben wir triviale und bekannte Resultate an.

2.3.1 Problemdefinitionen

Neben dem Formelauswertungsproblem betrachten wir noch drei weitere Probleme. Für das Erfüllbarkeitsproblem und das Tautologieproblem gibt es sowohl in

[20] Monotone Formeln enthalten als Operatoren nur $\wedge$ und $\vee$, aber keine Negation.

der Modallogik als auch in der intuitionistischen Logik bereits eine Reihe Resultate. Für einige Logiken können wir aber auch hier noch neue Resultate angeben. Das Modelläquivalenzproblem untersuchen wir nur am Rande.
Im weiteren Verlauf seien $\mathcal{F} \subseteq \mathfrak{F}$ ein beliebiges Formelfragment und $\mathcal{K} \subseteq \mathfrak{K}$ eine beliebige Modellklasse. Zuerst definieren wir das Formelauswertungsproblem.

Problem:	$\mathcal{F}$-$\mathcal{K}$-FA
Eingabe:	$\langle \alpha, \mathcal{M}, w \rangle$, wobei $\alpha \in \mathcal{F}$ eine Formel, $\mathcal{M} \in \mathcal{K}$ ein Modell und w eine Welt aus $\mathcal{M}$ ist.
Frage:	Gilt $\mathcal{M}, w \models \alpha$?

Das Erfüllbarkeitsproblem ist für eine gegebene Formel die Frage, ob es in $\mathcal{K}$ ein Modell gibt, das eine Welt hat, in der die Formel erfüllt ist.

Problem:	$\mathcal{F}$-$\mathcal{K}$-SAT
Eingabe:	$\langle \alpha \rangle$, wobei $\alpha \in \mathcal{F}$ eine Formel ist.
Frage:	Existiert ein Modell $\mathcal{M}$ mit einer Welt w, so dass $\mathcal{M}, w \models \alpha$ gilt?

Das Tautologieproblem ist die Frage, ob eine gegebene Formel eine $\mathcal{K}$-Tautologie ist.

Problem:	$\mathcal{F}$-$\mathcal{K}$-TAUT
Eingabe:	$\langle \alpha \rangle$, wobei $\alpha \in \mathcal{F}$ eine Formel ist.
Frage:	Ist α eine $\mathcal{K}$-Tautologie?

Diese drei Probleme sind die Standardprobleme bei Komplexitätsbetrachtungen von Logiken. Ein weiteres Problem ist die Frage, ob zwei Welten $\mathcal{F}$-äquivalent sind, es wird auch Modelläquivalenzproblem genannt.

Problem:	$\mathcal{F}$-$\mathcal{K}$-MÄQ
Eingabe:	$\langle \mathcal{M}_1, w_1, \mathcal{M}_2, w_2 \rangle$, wobei $\mathcal{M}_1, \mathcal{M}_2 \in \mathcal{K}$ Modelle sind und w_1 bzw. w_2 Welten aus $\mathcal{M}_1$ bzw. $\mathcal{M}_2$ sind.
Frage:	Gilt $(\mathcal{M}_1, w_1) \equiv_{\mathcal{F}} (\mathcal{M}_2, w_2)$?

2.3.2 Bekannte und einfache Resultate

Wir geben in diesem Abschnitt für verschiedene Logiken eine (unvollständige) Auswahl bekannter und einfacher Resultate für die oben definierten Probleme an. Dabei konzentrieren wir uns auf das Formelauswertungsproblem, das Erfüllbarkeitsproblem und das Tautologieproblem.

Vorher klären wir noch eine grundsätzliche Frage zur syntaktischen Überprüfung. Soll ein Algorithmus für eine Eingabe entscheiden, ob es sich um eine Ja-Instanz oder eine Nein-Instanz eines Problems handelt, muss er zusätzlich prüfen, ob die Eingabe überhaupt eine gültige Instanz dieses Problems ist. Das bedeutet, wenn zum Beispiel ein Modell aus einer Modellklasse Teil der Eingabe ist, muss der Algorithmus auch in der Lage sein, zu testen, ob dieses Modell überhaupt als Eingabe erlaubt ist. Um dies effektiv entscheiden zu können, ist es wichtig, dass die Modellklassen „sinnvoll“ (Definition 2.4) definiert sind.

Lemma 2.28 *Sei $\mathcal{K} \subseteq \mathfrak{K}$ eine Modellklasse. Das Problem, für eine gegebene Adjazenzmatrix zu entscheiden, ob das von ihr repräsentierte Modell in $\mathcal{K}$ liegt, ist in AC^0.*

Da die Modellklassen über prädikatenlogische Formeln definiert werden (Definition 2.4), folgt sofort, dass dieses Entscheidungsproblem in FO liegt. Man beachte, dass $\mathcal{K}$ nicht Bestandteil der Eingabe, sondern Teil der Problemdefinition ist. Nach Immerman [Imm99] gilt $\mathsf{FO} = \mathsf{AC}^0$ und damit Lemma 2.28.
Die Frage nach der syntaktischen Korrektheit muss auch für Formeln beantwortet werden, wenn sie Teil der Eingabe sind.

Lemma 2.29 *Sei $\mathcal{F} \subseteq \mathfrak{F}$ ein Formelfragment. Das Problem, für eine gegebene Formel $\alpha \in \mathfrak{F}$ zu entscheiden, ob $\alpha \in \mathcal{F}$, liegt in TC^0.*

Aus den Definitionen 2.1 und 2.2 kann man für jedes Formelfragment direkt eine kontextfreie Klammergrammatik ableiten. Für die Prüfung auf syntaktischen Korrektheit müssen im Wesentlichen die öffnenden und die schließenden Klammern gezählt. Dies ist in TC^0 möglich (siehe dazu [Vol99]) und damit folgt Lemma 2.29. Wesentlich für unserer Resultate ist, dass sich die syntaktische Korrektheit der Eingabe immer mit den Ressourcen von NC^1 entscheiden lässt. Da AC^0 und TC^0 in NC^1 enthalten sind, gehen wir bei weiteren Betrachtungen in dieser Arbeit immer von syntaktischer Korrektheit der Eingabe aus.

Das Formelauswertungsproblem

Das erste Resultat zeigt, dass wir bei dem Formelauswertungsproblem für die hier betrachteten Logiken keine Komplexitätsklassen oberhalb von P in Betracht ziehen müssen. Dieses triviale Resultat folgt aus [FL79].

Theorem 2.30 *Sei* L *eine Modallogik, dann ist das Problem* L-FA *in* P.

Beweis. Für eine Modallogik $\mathrm{L} = (\mathcal{F}, \mathcal{K})$ entscheidet Algorithmus 1 L-FA.
Die Korrektheit folgt direkt aus der Definition der Interpretation in der Modallogik (Definition 2.5). Für eine Instanz $\langle \alpha, (W, S, \xi, V), w \rangle$ ist die Laufzeit in $\mathcal{O}(|W|^2 \cdot |\mathrm{TF}(\alpha)|)$, also kann der Algorithmus das Problem in polynomieller Zeit entscheiden. □

Algorithmus 1 `Allgemeiner Formelauswerter`

Eingabe: Formel $\alpha \in \mathcal{F}$, Modell $\mathcal{M} = (W, S, \xi, V) \in \mathcal{K}$, Welt $w \in W$

A ist ein Booleanfeld mit Indizes aus $W \times \{1, 2, \ldots, |\mathrm{TF}(\alpha)|\}$
wiederhole für alle $i \in \{1, 2, \ldots, |\mathrm{TF}(\alpha)|\}$
  wiederhole für alle $u \in W$
    wenn $\alpha_i \in V$ **dann** $A(u, i) := u \in \xi(\alpha_i)$
    wenn $\alpha_i = \bot$ **dann** $A(u, i) :=$ `false`
    wenn $\alpha_i = \alpha_j \to \alpha_k$ **dann** $A(u, i) := A(u, j) \to A(u, k)$
    wenn $\alpha_i = \Box\alpha_j$ **dann**
      wenn $A(v, j)$ für alle $v \in W$ mit $(u, v) \in S$ **dann** $A(u, i) :=$ `true`
        sonst $A(u, i) :=$ `false`
wenn $A(w, \alpha)$ **dann** akzeptiere **sonst** lehne ab

Für eine weitere Klasse von Logiken folgt die Komplexität des Formelauswertungsproblems direkt aus bekannten Ergebnissen, da diese Logiken der Aussagenlogik sehr ähnlich sind.

Theorem 2.31 *Seien* $\mathfrak{K}_1^{\mathsf{i}} \subseteq \mathfrak{K}_{refl}^{\mathsf{i}}$ *die Menge aller intuitionistischen Modelle mit einer Welt, die sich selbst sieht und* $\mathrm{L}_1 = (\mathfrak{F}^{\mathsf{i}}, \mathfrak{K}_1^{\mathsf{i}})$*. Dann ist das Problem* L_1-FA *in* NC^1 *und* $\mathrm{L}_1[\bot, \rightarrowtriangle, 0]$-FA*,* $\mathrm{L}_1[\rightarrowtriangle, 1]$-FA*,* $\mathrm{L}_1[\bot, \wedge, \vee, 0]$-FA *und* $\mathrm{L}_1[\wedge, \vee, 1]$-FA *sind* NC^1*-hart.*

Beweis. Aus Definition 2.5 kann man direkt ableiten, dass die Auswertung einer Formel in Modellen bestehend aus einer Welt, die sich selbst sieht, genau dem Auswerten aussagenlogischer Formeln ohne Variablen entspricht. Die intuitionistische Implikation entspricht in diesen Modellen exakt der normalen Implikation. Jede Variable, die in der Formel vorkommt und in der Welt erfüllt ist, wird durch $\top$ ersetzt. Alle anderen Variablen ersetzt man durch $\bot$. Da $\mathrm{BFVP}[\wedge, \vee, \neg]$ in NC^1 ist [Bus87], folgt die obere Schranke sofort.
Auf gleiche Weise lassen sich die angegebenen unteren Schranken direkt aus den Theoremen 2.26 und 2.27 folgern. □

Beschränkt man in einer Modallogik die Operatoren auf die der Aussagenlogik, so ist für diese Logik das Problem der Formelauswertung ebenfalls in NC^1, da ohne modale Operatoren die Welt, in der ausgewertet werden soll, nicht verlassen werden kann. Das Problem ist wieder mit Hilfe von $\mathrm{BFVP}[\wedge, \vee, \neg]$ lösbar. Je nach Wahl der weiteren Einschränkungen kann das Problem dann natürlich auch NC^1-hart sein.
Das Formelauswertungsproblem ist für die allgemeine Modallogik K bereits ohne Verwendung von Variablen P-hart, wenn man die strikte Implikation ($\rightarrowtriangle$ bzw. im modalen Zusammenhang auch $\Box(\cdot \to \cdot)$) als einzigen Operator verwendet. Der Beweis von Theorem 2.32 illustriert die Verwendung der alternierenden Pfade, wie sie auch später in ähnlicher Form vorkommt.

Theorem 2.32 *Das Problem* K[$\bot, \rightarrow\!\!\triangleright, 0$]-FA *ist* P*-hart.*

Beweis. Wir zeigen zuerst, dass das Formelauswertungsproblem für K[1] P-hart ist. Dafür geben wir eine Reduktion von ASGEP auf K[1]-FA an. Die P-Härte folgt dann aus Theorem 2.21. Wir gehen diesen Umweg, da bei dieser Konstruktion der Zusammenhang zwischen dem alternierenden Pfad und der Formelauswertung sehr anschaulich ist. Für das eigentliche Ergebnis führen wir dann noch einige technische Modifikationen durch, mit denen wir die eine Variable und die Vorkommen von $\Diamond$ einsparen können.

Sei $\langle G, s, t\rangle$ eine ASGEP-Instanz, wobei $G = (V, E)$ ein alternierender Schichtgraph mit m Schichten ist (o.B.d.A. sei m gerade). Wir konstruieren $\mathcal{M}_G^1 = (W, S, \xi, \{p\})$ wie folgt. Der Rahmen von $\mathcal{M}_G^1$ ist genau der Graph G, also $W := V$ und $S := E$. Nur in den Zielknoten bzw. in die Zielwelt t setzen wir die Variable p mit $\xi(p) := \{t\}$. Zusätzlich definieren wir $\alpha_G^1 := \Diamond\Box\Diamond\Box \ldots \Diamond p$, wobei das Präfix $\Diamond\Box\Diamond\Box \ldots \Diamond$ aus $m-1$ alternierenden modalen Operatoren besteht. Jetzt gilt offensichtlich $\langle G, s, t\rangle \in$ ASGEP genau dann, wenn $\mathcal{M}_G^1, s \models \alpha_G^1$.

Zunächst wollen wir die Verwendung der Variablen p einsparen. Dafür modifizieren wir den Rahmen (W, S) derart, dass wir Kanten (w, w) für alle Welten $w \neq t$ aus der obersten Schicht V_m einfügen. Die so konstruierte Variante von $\mathcal{M}_G^1$ bezeichnen wir mit $\mathcal{M}_G$. In der obersten Schicht V_m von $\mathcal{M}_G$ ist t die einzige Welt, die keinen Nachfolger hat, und somit die einzige Welt, in der $\Box\bot$ erfüllt ist. Wir modifizieren α zu α', indem wir p durch $\Box\bot$ ersetzen. Damit gilt sofort $\langle G, s, t\rangle \in$ ASGEP genau dann, wenn $\mathcal{M}_G, s \models \alpha'$. Jetzt haben wir gezeigt, dass K[0]-FA P-hart ist.

Im letzten Schritt wollen wir uns auf die Verwendung von $\rightarrow\!\!\triangleright$ als einzigen Operator und $\bot$ als Konstante beschränken. Dafür geben wir eine neue Formel an. Mit Induktion über die Schichten analog zu der im Beweis von Behauptung 2.1 kann man zeigen, dass

$$\underbrace{\Box(\Box(\ldots\Box(}_{m-1}\ \Box(\top \rightarrow \bot)\ \underbrace{\rightarrow \bot) \rightarrow \bot) \cdots \rightarrow \bot)}_{m-1} \quad \equiv_{\mathfrak{K}} \quad \alpha'$$

gilt. Ersetzt man jetzt noch $\top$ durch $\Box(\bot \rightarrow \bot)$, so enthält die Formel nur die Konstante $\bot$ und die strikte Implikation. Wir setzen

$$\alpha_G \quad := \quad \underbrace{((\ldots(}_{m-1}\ ((\bot \rightarrow\!\!\triangleright\ \bot) \rightarrow\!\!\triangleright\ \bot)\ \underbrace{\rightarrow\!\!\triangleright\ \bot) \rightarrow\!\!\triangleright\ \bot) \ldots \rightarrow\!\!\triangleright\ \bot)}_{m-1}$$

und es gilt

$$\langle G, s, t\rangle \in \text{ASGEP} \quad \Longleftrightarrow \quad \mathcal{M}_G, s \models \alpha_G\ .$$

Mit Theorem 2.21 folgt die P-Härte von K[$\bot, \rightarrow\!\!\triangleright, 0$]-FA unmittelbar. □

Unsere weiteren Ergebnisse zum Formelauswertungsproblem sind in den folgenden Kapiteln zu finden.

Das Erfüllbarkeitsproblem

Das berühmteste Resultat ist Cooks Beweis von 1971 [Coo71b]. Er zeigte, dass das Erfüllbarkeitsproblem für die Aussagenlogik NP-vollständig ist. Dieses Resultat lässt sich auch auf das Erfüllbarkeitsproblem für die intuitionistischen Logiken mit reflexiven Modellen übertragen.

Theorem 2.33 *Die Probleme* IPL-SAT, KC-SAT, LC-SAT *sind* NP*-vollständig.*

Beweis. Wesentlich hierbei ist die Erkenntnis, dass die Menge der erfüllbaren Formeln aus allen Logiken des Theorems gleich der Menge der erfüllbaren Formeln der Aussagenlogik ist.
Für eine erfüllbare aussagenlogische Formel gibt es eine erfüllende Belegung. Aus dieser Belegung lässt sich ein intuitionistisches Kripke-Modell konstruieren. Dieses Modell besteht aus einer Welt, die sich selbst sieht. Jede Variable, die in der erfüllenden Belegung auf `true` gesetzt wird, ist in dieser Welt erfüllt. Alle anderen Variablen sind in der Welt nicht erfüllt. Betrachtet man die Formel jetzt als intuitionistische Formel, so wird sie offensichtlich von diesem Modell erfüllt. Die Interpretation von $\multimap$ und $\to$ ist in Welten, die keine echten Nachfolger haben, aber sich selbst sehen, identisch.
Umgekehrt gilt für erfüllbare intuitionistische Formeln, dass sie auch in einer Welt erfüllt sind, die keine echten Nachfolger hat, die also maximal ist. Abhängig davon, welche Variablen in dieser Welt gelten, kann man eine Belegung für die aussagenlogische Variante der Formel konstruieren. In dieser Welt haben $\multimap$ und $\to$ wieder dieselbe Bedeutung. Jede Variable, die in dieser maximalen Welt erfüllt ist, wird mit `true` belegt und alle anderen werden mit `false` belegt. Die so konstruierte Belegung erfüllt die Formel im aussagenlogischen Sinne. □

Aus dem Beweis wird ersichtlich, dass die NP-Härte auch dann noch gilt, wenn man die Logiken auf Modellklassen beschränkt, die nur Modelle mit einer (reflexiven) Welt enthalten. Für intuitionistische Logiken, deren Modelle nicht auf reflexiven Rahmen basieren, ist das Erfüllbarkeitsproblem deutlich einfacher.

Theorem 2.34 *Die Probleme* BPL-SAT *und* FPL-SAT *sind* NC^1*-vollständig.*

Beweis. Das Erfüllbarkeitsproblem für die monotone Aussagenlogik ist NC^1-hart [BM95]. Von diesem Problem kann man eine *cd*-Reduktion zu den Problemen aus dem Theorem angeben. Sei α eine monotone aussagenlogische Formel, in der die Variablen p_1 bis p_n vorkommen. Dann gilt offensichtlich, dass α genau dann erfüllbar ist, wenn $\alpha[p_1/\top][p_2/\top]\dots[p_n/\top]$ gültig ist. Außerdem gilt, dass $\alpha[p_1/\top]\dots[p_n/\top]$ genau dann gültig ist, wenn α in dem BPL-Modell (bzw. FPL-Modell) $\mathcal{M}_1 = (\{w\}, \varnothing, \xi, \mathrm{VAR})$ mit $\xi^{-1}(w) = \{p_1, p_2, \dots, p_n\}$ erfüllt ist. Wenn α BPL-erfüllbar (bzw. FPL-erfüllbar) ist, dann von diesem Modell.
Um zu testen, ob eine BPL-Formel (bzw. FPL-Formel) erfüllbar ist, ersetzt man zu erst alle $\multimap$-Teilformeln durch $\top$. In $\mathcal{M}_1$ sind diese alle erfüllt, weil w keine

Nachfolger hat. Übrig bleibt eine Formel, die nur $\wedge$ und $\vee$ enthält. In dieser werden alle atomaren Aussagen durch $\top$ ersetzt und es wird geprüft, ob die resultierende Formel aussagenlogisch gültig ist. Dies ist in NC^1 möglich [Bus87]. □

In der Modallogik ist das Erfüllbarkeitsproblem bereits umfangreich untersucht worden [Lad77, Spa93, HM92]. Das prominenteste Resultat stammt von Ladner.

Theorem 2.35 ([Lad77]) *Das Problem* K-SAT *ist* PSPACE*-vollständig.*

Auch für die Fragmente K4, S4 und PrL ist das Erfüllbarkeitsproblem PSPACE-vollständig [Lad77, Spa93]. Für S4.2 ist es NP-vollständig [BdV01, Lemma 6.40].

Das Tautologieproblem

In der Aussagenlogik sind das Tautologieproblem und das Erfüllbarkeitsproblem komplementär. Eine Formel ist genau dann eine Tautologie, wenn ihre Negation nicht erfüllbar ist. Damit folgt sofort, dass das Tautologieproblem für die Aussagenlogik coNP-vollständig ist. Da in der intuitionistischen Logik die Negation eine andere Bedeutung hat, gilt dieses Verhältnis zwischen Erfüllbarkeits- und Tautologieproblem nicht mehr. Obwohl $\neg(\alpha \vee \neg\alpha)$ nicht erfüllbar ist, ist $\alpha \vee \neg\alpha$ keine Tautologie. Dies ist wiederum eine Folgerung aus der Tatsache, dass der Satz des ausgeschlossenen Dritten in der intuitionistischen Logik nicht gilt. Das Tautologieproblem für IPL ist sogar noch schwerer als in der Aussagenlogik und dafür genügt bereits die intuitionistische Implikation als einziger Operator.

Theorem 2.36 ([Sta79, Cha85, Sve03b])
Das Problem IPL[$\rightarrow$]-TAUT *ist* PSPACE*-vollständig.*

Rybakov [Ryb06] konnte weitere Ergebnisse angeben, bei denen er sich auf Fragmente mit beschränkter Variablenzahl konzentriert.

Theorem 2.37 ([Ryb06]) *Die Probleme* BPL[0]-TAUT*,* FPL[1]-TAUT *und* IPL[2]-TAUT *sind* PSPACE*-vollständig.*

Das Resultat für IPL[2] lässt sich direkt auf die Logik KC[2] übertragen, da die Konstruktion aus dem Beweis von Rybakov [Ryb06, Lemma 7] analog für KC-Modelle funktioniert.
In der Modallogik K sind Tautologie- und Erfüllbarkeitsproblem wieder wie in der Aussagenlogik komplementär. Deswegen gilt für die allgemeine Modallogik K, dass das Tautologieproblem ebenfalls PSPACE-vollständig ist [Lad77]. Selbst die modale Variante, in der nur die strikte Implikation als Operator zu gelassen ist, ist noch PSPACE-vollständig [Bou04]. Beschränkt man die Zahl der Variablen, gilt für K und K4, dass das Tautologieproblem selbst für die Fragmente ohne Variablen PSPACE-vollständig ist [CR02]. Für S4 konnte in [CR02] gezeigt werden, dass auch das Fragment mit einer Variablen PSPACE vollständig ist. In [Sve03a] wurde auch für PrL mit einer Variablen die PSPACE-Vollständigkeit des Tautologieproblems gezeigt. Resultate zur PSPACE-Härte von modalen Fragmenten mit strikter Implikation und eine beschränkten Variablenzahl sind bisher noch nicht bekannt.

Kapitel 3

Endlich erzeugte Logiken

In diesem Kapitel betrachten wir die Formelauswertung für endlich erzeugte Logiken. Dabei bezeichnen wir eine intuitionistische[1] Logik $(\mathcal{F}, \mathcal{K})$ als *endlich erzeugt*, wenn $\mathcal{F}$ bezüglich $\equiv_{\mathcal{K}}$ in nur endlich viele Äquivalenzklassen zerfällt – es gibt also nur endlich viele Formeln, die paarweise nicht äquivalent sind. In der Literatur [dLHdJ12] wurde bereits für viele Logiken gezeigt, dass sie endlich erzeugt sind. Wir erweitern den Kreis dieser Logiken und zeigen für jedes $n \in \mathbb{N}$, dass die Logik LC[n] endlich erzeugt ist (Abschnitt 3.3). In einem späteren Kapitel zeigen wir außerdem, dass alle superintuitionistischen Logiken mit einer Variablen endlich erzeugt sind (Abschnitt 4.4).
Dieses Kapitel gliedert sich wie folgt: Abschnitt 3.1 beschäftigt sich mit dem Formelauswertungsproblem für endlich erzeugte Logiken. Wir zeigen, dass es immer in NC^1 liegt. In Abschnitt 3.2 geben wir einige Folgerungen an, die sich aus dem Beweis von Theorem 3.3 ergeben. Ganz konkrete Vertreter schauen wir uns in Abschnitt 3.3 an und zeigen für deren Formelauswertungsproblem auch die NC^1-Härte. Am Ende fassen wir die Ergebnisse dieses Kapitels in Abschnitt 3.4 kurz zusammen.

3.1 Formelauswertung in endlich erzeugten Logiken

Wir werden zeigen, dass für alle endlich erzeugten Logiken das Formelauswertungsproblem in NC^1 liegt. Dazu arbeiten wir mit sogenannten *Basen*. Eine Basis einer Logik ist eine Menge von Formeln, die aus jeder Äquivalenzklasse genau eine Formel enthält.

Definition 3.1 *Seien $\mathcal{F} \subseteq \mathfrak{F}^{\mathsf{i}}$ ein Formelfragment und $\mathcal{K} \subseteq \mathfrak{K}^{\mathsf{i}}$ eine Modellklasse. Wir bezeichnen die Logik $(\mathcal{F}, \mathcal{K})$ als endlich erzeugt, wenn es eine Menge $\{\varphi_1, \varphi_2, \ldots, \varphi_n\} \subseteq \mathcal{F}$ mit $n \in \mathbb{N}$ gibt, die folgende Eigenschaften erfüllt:*

(1) *$\varphi_i \not\equiv_{\mathcal{K}} \varphi_j$ für alle $i \neq j$ und $1 \leq i, j \leq n$.*

(2) *Für alle $\alpha \in \mathcal{F}$ gibt es ein $i \in \{1, 2, \ldots, n\}$ mit $\alpha \equiv_{\mathcal{K}} \varphi_i$.*

[1] Unser Haupttheorem gilt mit kleinen Modifikationen im Beweis auch für Modallogiken. Da aber die Vertreter in der Literatur und in dieser Arbeit alle intuitionistische Logiken sind, arbeiten wir in diesem Kapitel auch nur mit intuitionistischen Logiken.

Die Menge $\{\varphi_1, \varphi_2, \ldots, \varphi_n\}$ heißt Basis von $(\mathcal{F}, \mathcal{K})$ und die Formeln in einer Basis werden als Basisformeln bezeichnet.

Es ist offensichtlich, dass die Basis einer Logik nicht eindeutig bestimmt ist. Wir verwenden im weiteren Verlauf Basen, die möglichst kurze Formeln enthalten – sogenannte *normale* Basen. Aus Gründen der besseren Handhabung ordnen wir die Formeln in einer Basis nach ihrer Länge.

Definition 3.2 *Die Logik $(\mathcal{F}, \mathcal{K})$ sei endlich erzeugt und $B = \{\varphi_1, \varphi_2, \ldots, \varphi_n\}$ eine Basis von $(\mathcal{F}, \mathcal{K})$. Wir bezeichnen B als normale Basis, falls für alle Basisformeln folgende Eigenschaften gelten:*

(1) *Für alle $\alpha \in \mathcal{F}$ mit $\alpha \equiv_{\mathcal{K}} \varphi_i$ gilt $|\varphi_i| \leq |\alpha|$.*

(2) *Für alle $i < j \leq n$ gilt $|\varphi_i| \leq |\varphi_j|$.*

Für jede endlich erzeugte Logik $(\mathcal{F}, \mathcal{K})$ gibt es eine normale Basis, die nicht eindeutig bestimmt ist. Hat man eine beliebige Basis, kann jede Basisformel durch eine mit der kleinsten Länge aus ihrer Äquivalenzklasse ersetzt werden und schon hat man eine normale Basis aus einer beliebigen Basis konstruiert. Wir gehen o.B.d.A. davon aus, dass $(\mathcal{F}, \mathcal{K})$ nur paarweise nicht $\mathcal{K}$-äquivalente atomare Aussagen als Formeln enthält. Dann enthält eine normale Basis alle atomaren Aussagen aus $\mathcal{F}$. Aus der konstanten Größe der Basis ergibt sich, dass das Formelauswertungsproblem mit sehr wenigen Ressourcen gelöst werden kann. Statt eine gegebene Formel direkt in einer Welt auszuwerten, bestimmt man die äquivalente Basisformel und wertet diese dann in der Welt aus. Wir können jetzt das Haupttheorem dieses Abschnittes angeben.

Theorem 3.3 *Sei $(\mathcal{F}, \mathcal{K})$ eine endlich erzeugt Logik. Dann gilt $\mathcal{F}$-$\mathcal{K}$-FA $\in \mathsf{NC}^1$.*

Beweis. Im Beweis beschränken wir uns auf endlich erzeugte Logiken, die nur logische Operatoren aus $\{\wedge, \vee, \rightarrowtriangle\}$ enthalten. Kommen in der Logik andere Operatoren vor, die sich mit Hilfe von $\wedge$, $\vee$ und $\rightarrowtriangle$ darstellen lassen (z.B. $\neg$, $\neg\neg$ oder $\leftrightarrowtriangle$), verläuft der Beweis analog. Stellen, an denen auf die konkrete Definition der vorkommenden Operatoren Bezug genommen wird, lassen sich entsprechend der Konstruktion der Operatoren anpassen.
Es seien $(\mathcal{F}, \mathcal{K})$ eine endlich erzeugt Logik und $B = \{\varphi_1, \varphi_2, \ldots, \varphi_n\}$ eine normale Basis dieser Logik. Eine Instanz $\langle \alpha, \mathcal{M}, w \rangle$ des Formelauswertungsproblems wird wie folgt untersucht: Für die Formel α bestimmt man die $\mathcal{K}$-äquivalente Basisformel und in der Welt w wird dann nur noch diese Basisformel ausgewertet. Dafür definieren wir zunächst ein Entscheidungsproblem, bei dem gefragt wird, ob eine gegebene Formel äquivalent zu einer Basisformel ist. Wir kodieren dabei die Basisformel durch ihren Index in B.

Problem: $\mathcal{F}$-$\mathcal{K}$-B-BFÄQ

Eingabe: $\langle \alpha, i \rangle$, wobei $\alpha \in \mathcal{F}$ eine Formel und $i \in \{1, 2, \ldots, |B|\}$ ein Index ist.
Frage: Gilt $\alpha \equiv_{\mathcal{K}} \varphi_i$?

Als weiteres Problem definieren wir das Formelauswertungsproblem für $(\mathcal{F}, \mathcal{K})$ eingeschränkt auf die Basisformeln.

Problem: $\mathcal{F}$-$\mathcal{K}$-B-BFA
Eingabe: $\langle \varphi, \mathcal{M}, w \rangle$ wobei $\varphi \in B$ eine Basisformel, $\mathcal{M} \in \mathcal{K}$ ein Modell, und w ist eine Welt aus $\mathcal{M}$ ist.
Frage: Gilt $\mathcal{M}, w \models \varphi$?

Algorithmus 2 entscheidet das Formelauswertungsproblem für $(\mathcal{F}, \mathcal{K})$ nach dem oben beschriebenen Prinzip. Die Korrektheit folgt direkt aus den Eigenschaften einer (normalen) Basis. Die Komplexität analysieren wir im Folgenden mit Hilfe der Behauptungen 3.1 und 3.2. Wichtig hierbei ist, dass die Basis B weder berechnet werden muss, noch Teil der Eingabe ist, sie kann abhängig von der Logik einmal bestimmt werden und dann fest im Algorithmus implementiert werden.

Algorithmus 2 `Formelauswertung für die endlich erzeugte Logik` $(\mathcal{F}, \mathcal{K})$ `mit der Basis` $B = \{\varphi_1, \varphi_2, \ldots, \varphi_n\}$

Eingabe: eine Formel $\alpha \in \mathcal{F}$, ein Modell $\mathcal{M} \in \mathcal{K}$, und eine Welt w aus $\mathcal{M}$
1: $i := 0$
2: **wiederhole**
3: $\quad i := i + 1$
4: **bis** $\langle \alpha, i \rangle \in \mathcal{F}$-$\mathcal{K}$-$B$-BFÄQ
5: **wenn** $\langle \varphi_i, \mathcal{M}, w \rangle \in \mathcal{F}$-$\mathcal{K}$-$B$-BFA **dann** akzeptiere **sonst** lehne ab

Behauptung 3.1 *Das Problem* $\mathcal{F}$-$\mathcal{K}$-B-BFÄQ *ist in* NC^1.

Beweis der Behauptung. Sei $\varphi_i \in B$ eine Basisformel. Wir zeigen, dass die Menge $[\varphi_i]$ aller zu φ_i $\mathcal{K}$-äquivalenten Formeln aus $\mathcal{F}$ durch eine kontextfreie Klammersprache beschrieben werden kann. Wir geben für jede Basisformel φ_i eine kontextfreie Klammergrammatik G_{φ_i} so an, dass die Frage, ob eine gegebene Formel zu $[\varphi_i]$ gehört, gleich dem Wortproblem für die von G_{φ_i} erzeugte Sprache ist. Buss [Bus87] zeigte 1987, dass das Wortproblem für kontextfreie Klammersprachen in NC^1 ist. Da die Zahl der Basisformeln konstant ist, kann man für eine beliebige Formel und den Index einer Basisformel mit den Ressourcen von NC^1 bestimmen, ob die Formel in $[\varphi_i]$ ist.
Sei $\{q_1, q_2, \ldots, q_m\} = (\mathrm{VAR} \cup \{\bot, \top\}) \cap \mathcal{F}$ die Menge aller atomaren Aussagen und Konstanten von $\mathcal{F}$. Es gilt $m \leq n$ und ohne Einschränkungen gehen wir von $\varphi_1 = q_1$, $\varphi_2 = q_2$, $\ldots$, $\varphi_m = q_m$ aus. Des Weiteren seien $O \subseteq \{\wedge, \vee, \rightarrowtriangle\}$ die Menge der

in $\mathcal{F}$ vorkommenden logischen Operatoren, $\star \in O$ und $\alpha, \beta \in \mathcal{F}$ mit $\alpha \equiv_\mathcal{K} \varphi_j$ und $\beta \equiv_\mathcal{K} \varphi_k$ (wobei $\varphi_j, \varphi_k \in B$). Dann gibt es eine eindeutig bestimmte Basisformel $\varphi_{\langle\star,j,k\rangle}$, die nur von $\star$, j und k abhängt und für die $\varphi_{\langle\star,j,k\rangle} \equiv_\mathcal{K} \alpha\star\beta$ gilt. Dies ist eine direkte Folgerung aus Definition 3.1. Außerdem gilt $|\varphi_{\langle\star,j,k\rangle}| \leq |\varphi_j \star \varphi_k| \leq |\alpha \star \beta|$, da B eine normale Basis ist.
Wir geben jetzt für $1 \leq i \leq n$ eine kontextfreie Grammatik $G_{\varphi_i} = (N, T, P, S_i)$ an. Die von G_{φ_i} erzeugte Sprache ist $L(G_{\varphi_i})$ und wir zeigen $L(G_{\varphi_i}) = [\varphi_i] = \{\alpha \in \mathcal{F} \mid \alpha \equiv_\mathcal{K} \varphi_i\}$.

$$
\begin{aligned}
N &:= \{\varphi_1, \varphi_2, \ldots, \varphi_n\} \\
T &:= \{p_1, p_2, \ldots, p_m\} \cup O \cup \{(,)\} \\
P &:= \{\varphi_j \rightsquigarrow p_j \mid 1 \leq j \leq m\} \cup \\
&\quad \{\varphi_{\langle\star,j,k\rangle} \rightsquigarrow (\varphi_j \star \varphi_k) \mid j,k \in \{1,2,\ldots,n\},\ \star \in O,\ \varphi_{\langle\star,j,k\rangle} \equiv_\mathcal{K} \varphi_j \star \varphi_k\} \\
S_i &:= \varphi_i
\end{aligned}
$$

Seien $\alpha \in \mathcal{F}$ eine beliebige Formel und $\varphi \in B$ eine Basisformel. Wir beweisen, dass $\alpha \equiv_\mathcal{K} \varphi$ genau dann gilt, wenn $\alpha \in L(G_\varphi)$ ist.
Die Richtung von links nach rechts ($\alpha \equiv_\mathcal{K} \varphi \Rightarrow \alpha \in L(G_\varphi)$) zeigen wir mit vollständiger Induktion über α.

> Der Induktionsanfang für $\alpha \in \{q_1, q_2, \ldots, q_m\}$ ist klar. Für den Induktionsschritt sei $\alpha = \beta \star \gamma$ mit $\star \in O$. Dann gibt es Basisformeln $\varphi', \varphi'' \in B$ mit $\varphi' \equiv_\mathcal{K} \beta$ und $\varphi'' \equiv_\mathcal{K} \gamma$. Nach Induktionsvoraussetzung gilt $\beta \in L(G_{\varphi'})$ und $\gamma \in L(G_{\varphi''})$. Daher gibt es Ableitungen $\varphi' \rightsquigarrow \cdots \rightsquigarrow \beta$ bzw. $\varphi'' \rightsquigarrow \cdots \rightsquigarrow \gamma$ in $G_{\varphi'}$ bzw. $G_{\varphi''}$. Da $\alpha \equiv_\mathcal{K} \varphi$ und $\alpha = \beta\star\gamma$ ist, gilt $\varphi \equiv_\mathcal{K} \varphi'\star\varphi''$ und es gibt die Ableitungsregel $\varphi \rightsquigarrow \varphi' \star \varphi''$ in G_φ. Die Grammatiken G_φ, $G_{\varphi'}$ und $G_{\varphi''}$ unterscheiden sich nur in der Wahl des Startsymbols und haben insbesondere dieselben Regelmengen P. Deswegen sind $\varphi' \rightsquigarrow \cdots \rightsquigarrow \beta$ und $\varphi'' \rightsquigarrow \cdots \rightsquigarrow \gamma$ auch Ableitungen in G_φ. Damit ist auch $\varphi \rightsquigarrow \varphi'\star\varphi'' \rightsquigarrow \cdots \rightsquigarrow \beta\star\gamma = \alpha$ eine mögliche Ableitung in G_φ. Da φ das Startsymbol von G_φ ist (und α nur aus Terminalsymbolen besteht), gilt $\alpha \in L(G_\varphi)$.

Die andere Richtung ($\alpha \in L(G_\varphi) \Rightarrow \alpha \equiv_\mathcal{K} \varphi$) beweisen wir mit vollständiger Induktion über die Zahl der zur Ableitung von α verwendeten Regeln.

> Für die Ableitung von α benötigt man mindestens eine Regel. In diesem Fall ist $\alpha = q \in \{q_1, q_2, \ldots, q_m\}$ und die verwendete Regel ist $\varphi \rightsquigarrow q$. Nach Konstruktion gilt in diesem Fall $\alpha = \varphi$ und somit auch $\alpha \equiv_\mathcal{K} \varphi$. Für den Induktionsschritt nehmen wir an, dass es eine Ableitung der Form $\varphi \rightsquigarrow \cdots \rightsquigarrow \psi \rightsquigarrow \alpha$ gibt. Nach Induktionsvoraussetzung gilt $\varphi \equiv_\mathcal{K} \psi$. Aus der Konstruktion von G_φ (bzw. der Regeln aus P in der Grammatik) folgt, dass bei jeder Regel die linke Seite $\mathcal{K}$-äquivalent zur rechten Seite ist. Also gilt $\psi \equiv_\mathcal{K} \alpha$ und damit auch $\varphi \equiv_\mathcal{K} \alpha$.

Für eine feste Basisformel φ ist die Frage, ob eine gegebene Formel zu $[\varphi]$ gehört, äquivalent zum Wortproblem für $L(G_\varphi)$. Diese Frage kann mit den Ressourcen von NC^1 beantwortet werden [Bus87]. Da für eine $\mathcal{F}$-$\mathcal{K}$-B-BFÄQ-Instanz $\langle \alpha, i \rangle$

$$\alpha \in L(G_i) \iff \alpha \equiv_{\mathcal{K}} \varphi_i$$

gilt, ist $\mathcal{F}$-$\mathcal{K}$-B-BFÄQ $\in \mathsf{NC}^1$. ∎

Behauptung 3.2 *Das Problem $\mathcal{F}$-$\mathcal{K}$-B-BFA ist in AC^0.*

Beweis der Behauptung. Wir zeigen, dass $\mathcal{F}$-$\mathcal{K}$-B-BFA in FO ist. 1999 zeigte Immerman [Imm99, Theorem 5.22], dass $\mathsf{FO} = \mathsf{AC}^0$ gilt.
Wir konstruieren eine prädikatenlogische Formel Φ, die nur von $\mathcal{F}$ und $\mathcal{K}$ (bzw. der gewählten normalen Basis $B = \{\varphi_1, \varphi_2, \ldots, \varphi_n\}$) abhängt. Jede Instanz von $\mathcal{F}$-$\mathcal{K}$-B-BFA ist eine zu Φ passende Struktur und Φ wird von einer solchen Struktur genau dann erfüllt, wenn es sich bei der entsprechenden Instanz um eine Ja-Instanz handelt.
Es seien $\mathcal{M} = (W, S, \xi, V)$, $\mathcal{M} \in \mathcal{K}$ und $w \in W$ eine Welt aus $\mathcal{M}$. In einem ersten Schritt geben wir Formeln Φ_i für $1 \leq i \leq n$ so an, dass $\Phi_i(\mathcal{M}, w)$ genau dann wahr ist, wenn $\mathcal{M}, w \models \varphi_i$ gilt. O.B.d.A. nehmen wir wieder $\{q_1, q_2, \ldots, q_m\} = (\mathrm{VAR} \cup \{\bot, \top\}) \cap \mathcal{F}$ mit $\varphi_1 = q_1$, $\varphi_2 = q_2$, …, $\varphi_m = q_m$ an $(m \leq n)$. Damit ergeben sich die ersten m Formeln wie folgt:

$$\begin{aligned}
\Phi_1(\mathcal{M}, w) &:= w \in \xi(q_1) \\
\Phi_2(\mathcal{M}, w) &:= w \in \xi(q_2) \\
&\vdots \\
\Phi_m(\mathcal{M}, w) &:= w \in \xi(q_m)
\end{aligned}$$

Für $i > m$ besteht φ_i aus mehreren Teilformeln. Es sei $\varphi_i = \alpha \star \beta$ mit $\alpha \equiv_{\mathcal{K}} \varphi_j$, $\beta \equiv_{\mathcal{K}} \varphi_k$ und $\star \in \{\wedge, \vee, \twoheadrightarrow\}$. Die Konstruktion von Φ_i hängt von $\star$ ab.

$$\Phi_i(\mathcal{M}, w) := \begin{cases} \Phi_j(\mathcal{M}, w) \wedge \Phi_k(\mathcal{M}, w), & \text{falls } \star = \wedge \\ \Phi_j(\mathcal{M}, w) \vee \Phi_k(\mathcal{M}, w), & \text{falls } \star = \vee \\ \forall v \ : \ (w, v) \in S \to (\Phi_j(\mathcal{M}, v), \to \Phi_k(\mathcal{M}, v)), & \text{falls } \star = \twoheadrightarrow \end{cases}$$

Kommt Φ_j in Φ_i als Teilformel vor, dann bedeutet das, dass φ_i eine zu φ_j $\mathcal{K}$-äquivalente Teilformel α enthält. Da die Basis normal ist, gibt es keine Formel, die zu α $\mathcal{K}$-äquivalent und kürzer als α ist, da sonst φ_i nicht die kürzeste Formel der Klasse $[\varphi_i]$ wäre. Es folgt $|\varphi_j| = |\alpha| < |\varphi_i|$ und damit gilt $j < i$. Die Formeln Φ_i sind also alle endlich groß und hängen nur von $(\mathcal{F}, \mathcal{K})$ (bzw. der gewählten normalen Basis B) ab. Mit einer einfachen Induktion über den Formelaufbau von φ_i kann man zeigen, dass für alle $i \in \{1, 2, \ldots, n\}$, alle Modelle $\mathcal{M} \in \mathcal{K}$ und alle Welten w von $\mathcal{M}$ die Formel $\Phi_i(\mathcal{M}, w)$ genau dann wahr ist, wenn $\mathcal{M}, w \models \varphi_i$ gilt.

Als nächstes konstruieren wir Φ auf Basis der Φ_i.

$$\Phi(\varphi_i, \mathcal{M}, w) \quad := \bigvee_{k=1,2,\ldots,n} (\varphi_i = \varphi_k \wedge \Phi_k(\mathcal{M}, w))$$

Die Korrektheit dieser Konstruktion folgt direkt aus der der Φ_i-Formeln. Jetzt ist noch zu klären, ob $\langle \varphi_i, \mathcal{M} = (W, S, \xi, V), w \rangle$ eine passende Struktur ist. Wir fassen die Instanz wie folgt auf: Das Universum U besteht aus allen Welten des Modells und den Basisformeln der gewählten Basis B – also $U = W \cup B$. Daher ist zu beachten, dass jede atomare Aussage und jede Konstante, die in Formeln aus $\mathcal{F}$ vorkommt, auch in B vertreten ist.

- W und V sind einstellige Prädikate über U,
- S ist ein zweistelliges Prädikat über U,
- ξ ist ein zweistelliges Prädikat über U (bzw. zwischen $\mathrm{VAR} \cap \mathcal{F}$ und W) und
- φ_i und w sind Individualkonstanten.

In diesem Sinne ist $\langle \varphi_i, \mathcal{M} = (W, S, \xi, V), w \rangle$ eine zu Φ passende, endliche Struktur und es folgt $\mathcal{F}$-$\mathcal{K}$-B-BFA $\in \mathsf{FO}$. Mit Immerman [Imm99, Theorem 5.22] gilt $\mathcal{F}$-$\mathcal{K}$-B-BFA $\in \mathsf{AC^0}$. ▪

Mit diesen beiden Ergebnissen können wir jetzt die Komplexität von Algorithmus 2 analysieren. Als Eingabe bekommt der Algorithmus eine Instanz $\langle \alpha, \mathcal{M}, w \rangle$ und durchläuft die Schleife von Zeile 2 bis 4 höchstens n mal. Dabei ist n – die Anzahl der Basisformeln in B – konstant und unabhängig von der Eingabe, da n nur von der Logik und nicht der konkreten Eingabe abhängt. Aus Behauptung 3.1 folgt, dass die Abbruchbedingung in Zeile 4 mit den Ressourcen von $\mathsf{NC^1}$ überprüft werden kann. Dass die Entscheidung in Zeile 5 mit den Ressourcen von $\mathsf{AC^0}$ möglich ist, folgt aus der Behauptung 3.2. Da $\mathsf{AC^0} \subset \mathsf{NC^1}$ gilt [Smo87], kommt Algorithmus 2 mit den Ressourcen von $\mathsf{NC^1}$ aus und damit gilt $\mathcal{F}$-$\mathcal{K}$-FA $\in \mathsf{NC^1}$. □

Für eine konkrete endlich erzeugte Logik hängt die untere Schranke natürlich von weiteren Eigenschaften ab. In unserem Fall sind Logiken meist dann auch $\mathsf{NC^1}$-hart, wenn sie die Aussagenlogik ohne Variablen als Fragment enthalten.

3.2 Nebenresultate

Die Behauptungen 3.1 und 3.2 aus dem Beweis von Theorem 3.3 liefern einige interessante Nebenresultate, die wir hier kurz zusammenfassen wollen.

Lemma 3.4 *Sei* $(\mathcal{F}, \mathcal{K})$ *eine endlich erzeugte Logik. Dann sind die Probleme* $\mathcal{F}$-$\mathcal{K}$-SAT *und* $\mathcal{F}$-$\mathcal{K}$-TAUT *in* $\mathsf{NC^1}$.

Beweis. Diese beiden Aussagen können als Folgerung von Behauptung 3.1 gesehen werden. Seien $(\mathcal{F}, \mathcal{K})$ eine endlich erzeugte Logik und $B = \{\varphi_1, \varphi_2, \ldots, \varphi_n\}$ eine normale Basis von $(\mathcal{F}, \mathcal{K})$. Algorithmus 3 entscheidet die $\mathcal{K}$-Äquivalenz von zwei Formeln aus $\mathcal{F}$. Da n konstant und unabhängig von der Eingabe ist und $\mathcal{F}$-$\mathcal{K}$-B-BFÄQ $\in \mathsf{NC}^1$ gilt (Behauptung 3.1), kann diese Entscheidung in NC^1 getroffen werden.

Algorithmus 3 `Formeläquivalenz für die endlich erzeugte Logik` $(\mathcal{F}, \mathcal{K})$ `mit der Basis` $\{\varphi_1, \varphi_2, \ldots, \varphi_n\}$

Eingabe: Formeln $\alpha, \beta \in \mathcal{F}$

1: *equal* := `false`
2: **wiederhole für** $i := 1$ **bis** n
3: **wenn** $\langle \alpha, i \rangle \in \mathcal{F}$-$\mathcal{K}$-$B$-BFÄQ **und** $\langle \beta, i \rangle \in \mathcal{F}$-$\mathcal{K}$-$B$-BFÄQ
4: **dann** *equal* := `true`
5: **wenn** *equal* = `true` **dann** akzeptiere **sonst** lehne ab

$\mathcal{F}$-$\mathcal{K}$-SAT und $\mathcal{F}$-$\mathcal{K}$-TAUT sind spezielle Versionen der Frage, ob zwei Formeln äquivalent sind. Für eine Formel $\alpha \in \mathcal{F}$ gilt genau dann $\langle \alpha \rangle \in \mathcal{F}$-$\mathcal{K}$-SAT, wenn α und $\bot$ nicht äquivalent sind. Ebenso gilt genau dann $\langle \alpha \rangle \in \mathcal{F}$-$\mathcal{K}$-TAUT, wenn α äquivalent zu $\top$ ist. Da NC^1 unter Komplementierung abgeschlossen ist, sind $\mathcal{F}$-$\mathcal{K}$-SAT und $\mathcal{F}$-$\mathcal{K}$-TAUT in NC^1. □

Auch für die Modelle endlich erzeugter Logiken ergibt sich ein Resultat direkt aus Behauptung 3.2.

Lemma 3.5 *Sei $(\mathcal{F}, \mathcal{K})$ eine endlich erzeugte Logik. Dann ist das Problem $\mathcal{F}$-$\mathcal{K}$-MÄQ in AC^0.*

Beweis. Es seien $(\mathcal{F}, \mathcal{K})$ eine endlich erzeugte Logik und $B = \{\varphi_1, \varphi_2, \ldots, \varphi_n\}$ eine normale Basis. Wir zeigen $\mathcal{F}$-$\mathcal{K}$-MÄQ $\in \mathsf{FO}$. Mit Immerman [Imm99, Theorem 5.22] folgt $\mathcal{F}$-$\mathcal{K}$-MÄQ $\in \mathsf{AC}^0$. Für eine $\mathcal{F}$-$\mathcal{K}$-MÄQ-Instanz $\langle (\mathcal{M}_1, w_1), (\mathcal{M}_2, w_2) \rangle$ konstruieren wir eine prädikatenlogische Formel Ψ, die genau dann unter der Struktur $\langle (\mathcal{M}_1, w_1), (\mathcal{M}_2, w_2) \rangle$ wahr ist, wenn $(\mathcal{M}_1, w_1) \equiv_{\mathcal{F}} (\mathcal{M}_2, w_2)$ gilt. Dazu sei Φ die in Beweis von Behauptung 3.2 konstruierte Formel:

$$\Psi((\mathcal{M}_1, w_1), (\mathcal{M}_2, w_2)) \quad := \quad \bigwedge_{k=1,2,\ldots,n} (\Phi(\varphi_k, \mathcal{M}_1, w_1) \leftrightarrow \Phi(\varphi_k, \mathcal{M}_2, w_2)) \ .$$

Dabei ist n – die Größe der Basis – wieder nur abhängig von der Logik und nicht von der Eingabeinstanz. Offensichtlich erfüllt Ψ die geforderte Eigenschaft und es folgt $\mathcal{F}$-$\mathcal{K}$-MÄQ $\in \mathsf{FO}$ bzw. $\mathcal{F}$-$\mathcal{K}$-MÄQ $\in \mathsf{AC}^0$. □

Eine weitere naheliegende Beobachtung ist, dass es in einer endlich erzeugten Logik auch nur endlich viele verschiedene Modelle gibt.

Lemma 3.6 *Sei $(\mathcal{F}, \mathcal{K})$ eine Logik. Dann gilt, $(\mathcal{F}, \mathcal{K})$ ist genau dann endlich erzeugt, wenn es bezüglich $\equiv_{\mathcal{F}}$ nur endlich viele Äquivalenzklassen gibt.*

Beweis. Es seien $(\mathcal{F}, \mathcal{K})$ endlich erzeugt und B eine Basis. Zwei Welten sind nach Definition 2.7 $\mathcal{F}$-äquivalent, wenn sie genau dieselben Formeln aus $\mathcal{F}$ erfüllen. Genau genommen genügt es dabei, sich auf je einen Repräsentanten der Formeläquivalenzklassen bezüglich $\equiv_{\mathcal{K}}$ zu beziehen, anstatt alle Formeln zu betrachten. Es reicht also aus, die Basisformeln zu untersuchen. Da es nur konstant viele Basisformeln in B gibt, kann es auch nur endlich viele verschiedene Möglichkeiten geben, welche Basisformeln in einer Welt erfüllt sind und welche nicht. Konkret gibt es also maximal $2^{|B|}$ viele Äquivalenzklassen bezüglich $\equiv_{\mathcal{F}}$.
Sei umgekehrt $(\mathcal{F}, \mathcal{K})$ eine Logik mit endlich vielen Äquivalenzklassen bezüglich $\equiv_{\mathcal{F}}$. Analog zu einer Basis kann man eine Menge B' angeben, die aus jeder Äquivalenzklasse bezüglich $\equiv_{\mathcal{F}}$ genau einen Repräsentanten enthält – eine Art Modellbasis. Zwei Formeln sind $\mathcal{K}$-äquivalent, wenn sie in jedem dieser Repräsentanten gleich ausgewertet werden. Also kann es nur maximal $2^{|B'|}$ viele Formeln geben, die paarweise nicht $\mathcal{K}$-äquivalent sind. Daraus folgt, dass $(\mathcal{F}, \mathcal{K})$ endlich erzeugt ist. □

3.3 Einige Vertreter

Wir wollen in diesem Abschnitt einige endlich erzeugte Logiken angeben. Eine intuitionistische Logik ist genau dann endlich erzeugt, wenn die korrespondierende Heyting-Algebra endlich ist. Anschaulich bedeutet dies, dass die Elemente der Grundmenge einer Heyting-Algebra gerade den Formeläquivalenzklassen der zugehörigen Logik entsprechen (siehe dazu [vDT88, vD04]). Zuerst geben wir einige Resultate aus der Literatur an und dann zeigen wir, dass die Fragmente von LC mit einer beschränkten Variablenzahl endlich erzeugt sind.

3.3.1 Bekannte endlich erzeugte Logiken

Intuitionistische Logiken mit einer konstanten Zahl an Variablen und ohne Disjunktion sind endlich erzeugt. Bei der Untersuchung dieser Logiken wurde immer davon ausgegangen, dass die Konstante $\top$ Bestandteil der Formeln ist. Erste Resultate dazu gab es von Skolem [Sko53] und Balbes [Bal73]. Sie zeigten, dass $\mathrm{IPL}[O, n]$ für $n \in \{0, 1, 2\}$ und $\{\top, \rightarrowtriangle\} \subseteq O \subseteq \{\top, \wedge, \rightarrowtriangle\}$ endlich erzeugt sind. Dieses Resultat wurde von McKay [McK68] und Urquhart [Urq74] verallgemeinert.

Theorem 3.7 ([McK68, Urq74]) *Die Logiken*

$\mathrm{IPL}[O, n]$ *für* $n \in \mathbb{N}$ *und* $\{\top, \rightarrowtriangle\} \subseteq O \subseteq \{\top, \wedge, \rightarrowtriangle\}$

sind endlich erzeugt.

Ist $\bot$ kein Bestandteil der Logik, kann man die intuitionistische Negation $\neg$ nicht mit der intuitionistischen Implikation $\rightarrowtriangle$ konstruieren. In [dLHdJ12] wurden die Resultate weiter verallgemeinert und auch Logiken mit Negation untersucht.

Theorem 3.8 ([dLHdJ12]) *Die Logiken*

$$\begin{aligned} \mathrm{IPL}[O,n] \quad & \textit{für } n \in \mathbb{N} \textit{ und } \{\top, \rightarrowtriangle\} \subseteq O \subseteq \{\top, \wedge, \neg, \rightarrowtriangle\} \textit{ und} \\ & \textit{für } n \in \mathbb{N} \textit{ und } \{\top, \rightarrowtriangle\} \subseteq O \subseteq \{\top, \wedge, \neg\neg, \rightarrowtriangle\} \end{aligned}$$

sind endlich erzeugt.

Für das Formelauswertungsproblem ergeben sich mit den Theoremen 3.3 und 2.31 folgende Resultate.

Theorem 3.9 *Für alle Logiken aus den Theoremen 3.7 und 3.8 ist das Formelauswertungsproblem in* NC^1*. Enthalten die Logiken Variablen oder lässt sich die Konstante* $\bot$ *konstruieren, so ist ihr Formelauswertungsproblem auch* NC^1*-hart. In den anderen Fällen ist es trivial.*

In [dLHdJ12] wird erwähnt, dass einige Logiken mit Disjunktion und Negation, aber ohne Implikation ebenfalls endlich erzeugt sind. Leider gibt es dazu keine Beweise, sondern nur den Hinweis, dass diese Logiken in einer zukünftigen Arbeit untersucht werden sollen. Es handelt sich dabei um die Logiken $(\mathfrak{F}^{\mathsf{i}}[\neg, \wedge, n], \mathfrak{K}^{\mathsf{i}}_{refl}[n])$, $(\mathfrak{F}^{\mathsf{i}}[\neg(\cdot \vee \cdot), n], \mathfrak{K}^{\mathsf{i}}_{refl}[n])$, $(\mathfrak{F}^{\mathsf{i}}[\neg\neg(\cdot \wedge \cdot), \vee, n], \mathfrak{K}^{\mathsf{i}}_{refl}[n])$ und $(\mathfrak{F}^{\mathsf{i}}[\neg\neg, \vee, n], \mathfrak{K}^{\mathsf{i}}_{refl}[n])$ für $n \geq 1$.
Jedes Fragment der Aussagenlogik mit einer beschränkten Zahl von Variablen ist zwar auch endlich erzeugt, aber dass das Formelauswertungsproblem für diese Logiken in NC^1 liegt, folgt bereits aus Theorem 2.26 bzw. [Bus87].

3.3.2 LC mit beschränkter Variablenzahl

Die Logik LC ist eine intuitionistische Logik, bei der nur Modelle zugelassen sind, deren Rahmen eine lineare Ordnung bilden. Wir betrachten jetzt Fragmente von LC mit beschränkter Variablenzahl – die Logiken LC$[n]$ mit $n \in \mathbb{N}$.

Theorem 3.10 *Für alle* $n \in \mathbb{N}$ *ist* LC$[n]$ *endlich erzeugt.*

Beweis. Für jedes $n \in \mathbb{N}$ zeigen wir, dass es nur endlich viele verschiedene LC$[n]$-Modelle (also Äquivalenzklassen bezüglich $\equiv_{\mathfrak{F}^{\mathsf{i}}[n]}$) gibt. Daraus folgt mit Lemma 3.6, dass LC$[n]$ endlich erzeugt ist.
Wir definieren eine Modellklasse $\mathcal{K}^n$, die nur aus endlich vielen Modellen besteht. Dabei gilt, dass es für jede Welt aus jedem LC$[n]$-Modell eine $\mathfrak{F}^{\mathsf{i}}[n]$-äquivalente Welt in einem Modell aus $\mathcal{K}^n$ gibt. Ein Modell $\mathcal{M}' = (V, \preceq', \xi', \mathrm{VAR}_n)$ ist in $\mathcal{K}^n$, wenn Folgendes gilt:

$$\begin{aligned} V &= \{v_1, v_2, \ldots, v_m\} \text{ mit } 1 \leq m \leq n+1 \ , \\ \preceq' &= \{(v_a, v_b) \mid v_a, v_b \in V \text{ und } a \leq b\} \ , \\ \xi' &\text{ beliebig mit } \xi'^{-1}(v_a) \subsetneq \xi'^{-1}(v_b) \text{ für alle } 0 \leq a < b \leq n \ . \end{aligned}$$

Die Modelle aus $\mathcal{K}^n$ bestehen aus höchstens $n+1$ Welten. Paarweise verschiedene Welten in einem Modell haben unterschiedliche Belegungen. Es ist offensichtlich, dass $\mathcal{K}^n$ nur endlich viele Modelle enthält, da jedes Modell höchstens $n+1$ Welten haben kann und es für jede Welt nur maximal 2^n verschiedene Belegungen gibt. Wir zeigen jetzt, dass es für jede Welt w in jedem LC[n]-Modell $\mathcal{M}$ eine Welt w' in einem $\mathcal{K}^n$-Modell $\mathcal{M}'$ mit $(\mathcal{M}, w) \equiv_{\mathfrak{F}^i[n]} (\mathcal{M}', w')$ gibt.
Es sei $\mathcal{M} = (W, \preceq, \xi, \mathrm{VAR}_n)$ ein LC[n]-Modell mit $|W| = k$ und für den Rahmen $(W, \preceq)$ gilt

$$\begin{aligned} W &= \{w_1, w_2, \ldots, w_k\} \ , \\ w_a \preceq w_b &\Leftrightarrow a \leq b \text{ für } 0 \leq a, b \leq k \ . \end{aligned}$$

Wir definieren jetzt eine Funktion $f : W \mapsto \{v_1, v_2, \ldots, v_{n+1}\}$, die Welten aus W abhängig von ihrer Belegung Welten aus $\{v_1, v_2, \ldots, v_{n+1}\}$ zuordnet. Für jede Belegung, die in einer Welt aus $\mathcal{M}$ vorkommt, soll es in dem zugehörigen Modell $\mathcal{M}'$ genau eine Welt geben. Dafür wählen wir $W' = \{w_{i_1}, w_{i_2}, \ldots, w_{i_j}\} \subseteq W$ mit folgenden Eigenschaften:

(1) Für alle $w_{i_x}, w_{i_y} \in W'$ gilt $x \leq y \Leftrightarrow w_{i_x} \preceq w_{i_y}$.

(2) Für alle $w \in W$ gibt es ein $w' \in W'$ mit $\xi^{-1}(w) = \xi^{-1}(w')$.

(3) Für alle $w \in W, w' \in W'$ gilt $(w \preceq w' \ \& \ w \neq w') \Rightarrow \xi^{-1}(w) \subsetneq \xi^{-1}(w')$.

Anschaulich gesprochen ist eine Welt $w_{i_\ell} \in W'$ immer die kleinste Welt aus W (bezüglich $\preceq$) mit der Belegung $\xi^{-1}(w_{i_\ell})$. In allen echten Vorgängern von w_{i_ℓ} in W sind weniger atomare Aussagen erfüllt. Aus der Monotonie von ξ folgt direkt $|W'| \leq n+1$. Seien $w \in W$ und $w_{i_\ell} \in W'$ mit $\xi^{-1}(w) = \xi^{-1}(w_{i_\ell})$, dann setzen wir

$$f(w) := v_\ell \ .$$

Haben zwei Welten $w, w' \in W$ dieselbe Belegung ($\xi^{-1}(w) = \xi^{-1}(w')$), so werden sie mittels f derselben Welt aus $\{v_1, v_2, \ldots, v_{n+1}\}$ zu geordnet. Mit Hilfe dieser Zuordnung f geben wir nun das zu $\mathcal{M}$ gehörende Modell $\mathcal{M}' = (V, \preceq', \xi', \mathrm{VAR}_n)$ aus $\mathcal{K}^n$ an:

$$\begin{aligned} V &:= f(W) \ , \\ \preceq' &:= \{(v, v') \mid \exists w, w' \in W : f(w) = v, f(w') = v' \text{ und } w \preceq w'\} \ , \\ \xi'(p_i) &:= f(\xi(p_i)) \qquad \text{für } 1 \leq i \leq n \ . \end{aligned}$$

Diese Konstruktion wird durch ein Beispiel in Abbildung 3.1 illustriert. Zur Vereinfachung sagen wir, dass $\mathcal{M}'$ mit Hilfe von f aus $\mathcal{M}$ erzeugt wurde und schreiben $f(\mathcal{M}) = \mathcal{M}'$. Offensichtlich ist $\mathcal{M}'$ ein lineares Modell aus $\mathcal{K}^n$.

Behauptung 3.3 *Für alle* LC[n]*-Modelle* $\mathcal{M}$ *und alle Welten* w *aus* $\mathcal{M}$ *gilt* $(\mathcal{M}, w) \equiv_{\mathfrak{F}^i[n]} (f(\mathcal{M}), f(w))$.

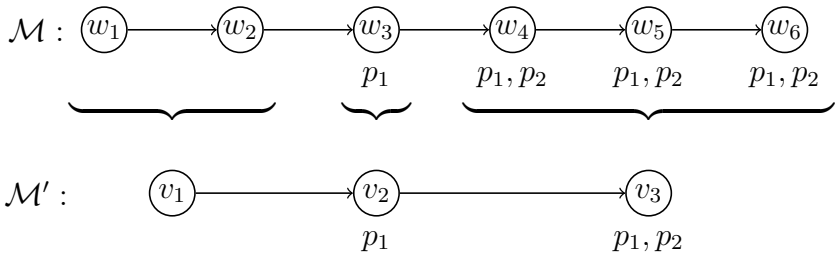

Abbildung 3.1: An einem Beispiel ist hier zu sehen, wie aus einem LC[n]-Modell $\mathcal{M}$ ein $\mathcal{K}^n$-Modell $\mathcal{M}'$ konstruiert wird. Alle Welten aus $\mathcal{M}$, in denen dieselben atomaren Aussagen erfüllt sind, werden zu einer Welt in $\mathcal{M}'$ zusammengefasst. Transitive und reflexive Kanten sind nicht abgebildet.

Beweis der Behauptung. Seien $\mathcal{M}$ ein LC[n]-Modell und w eine Welt aus $\mathcal{M}$. Wir zeigen, dass für alle $\alpha \in \mathfrak{F}^{\mathrm{i}}[n]$ genau dann $\mathcal{M}, w \models \alpha$ gilt, wenn auch $f(\mathcal{M}), f(w) \models \alpha$ gilt. Den Beweis führen wir mit vollständiger Induktion über den Formelaufbau von α. Sei $\mathcal{M} = (W, \preceq, \xi, \mathrm{VAR}_n)$ und $f(\mathcal{M}) = (V, \preceq', \xi', \mathrm{VAR}_n)$. Der Induktionsanfang ist für $\alpha = \bot$ klar. Für $\alpha \in \mathrm{VAR}_n$ folgt die Behauptung direkt aus der Konstruktion von f. Für den Induktionsschritt sei $\alpha = \beta \star \gamma$ mit $\star \in \{\wedge, \vee, \twoheadrightarrow\}$. Die Fälle $\star = \wedge$ und $\star = \vee$ sind einfach, die Behauptung folgt direkt aus der Induktionsvoraussetzung. Für $\star = \twoheadrightarrow$ gelten folgende Äquivalenzen:

$$
\begin{aligned}
&\mathcal{M}, w \models \beta \twoheadrightarrow \gamma && \text{(i)}\\
\Leftrightarrow\quad &\forall u \in W, w \preceq u : \mathcal{M}, u \models \beta \Rightarrow \mathcal{M}, u \models \gamma && \text{(ii)}\\
\Leftrightarrow\quad &\forall u \in W, w \preceq u : f(\mathcal{M}), f(u) \models \beta \Rightarrow f(\mathcal{M}), f(u) \models \gamma && \text{(iii)}\\
\Leftrightarrow\quad &\forall v \in V, f(w) \preceq' v : f(\mathcal{M}), v \models \beta \Rightarrow f(\mathcal{M}), v \models \gamma && \text{(iv)}\\
\Leftrightarrow\quad &f(\mathcal{M}), f(w) \models \beta \twoheadrightarrow \gamma && \text{(v)}
\end{aligned}
$$

Die Äquivalenz zwischen (i) und (ii) folgt aus der Definition der Interpretation von $\twoheadrightarrow$. Die Induktionsvoraussetzung liefert die Äquivalenz zwischen (ii) und (iii). Aus der Konstruktion von f folgt sofort (iii) $\Rightarrow$ (iv). Um (iv) $\Rightarrow$ (iii) zu zeigen, nehme man an, (iv) gilt und es gibt ein $u \in W$ mit $w \preceq u$ und $\mathcal{M}, u \models \beta$, aber $\mathcal{M}, u \not\models \gamma$. Für diese Welt u gilt dann aber auch $f(\mathcal{M}), f(u) \models \beta$ und $f(\mathcal{M}), f(u) \not\models \gamma$ und $f(w) \preceq' f(u)$. Dies ist ein Widerspruch zu (iv). Die Äquivalenz zwischen (iv) und (v) folgt wieder direkt aus der Definition der Interpretation von $\twoheadrightarrow$. ∎

Aus Behauptung 3.3 folgt, dass es für jede Welt w aus jedem LC[n]-Modell $\mathcal{M}$ eine Welt w' aus einem Modell $\mathcal{M}' \in \mathcal{K}^n$ gibt, so dass $(\mathcal{M}, w) \equiv_{\mathfrak{F}^{\mathrm{i}}[n]} (\mathcal{M}', w')$ gilt. Da es nur endlich viele Modelle in $\mathcal{K}^n$ gibt, folgt mit Lemma 3.6, dass $\mathfrak{F}^{\mathrm{i}}[n]$ endlich erzeugt ist. □

Anschaulich gesehen kann in einem LC[n]-Modell zwischen Welten, die eine identische Belegung aufweisen, nicht unterschieden werden. Deswegen muss jede Belegung, die in einem Modell vorkommt, nur einmal betrachtet werden. Da es nur endlich viele verschiedene Belegungen gibt, sind auch nur endlich viele verschiedene Modelle möglich.
Für das Formelauswertungsproblem der LC[n]-Logiken können wir folgende untere Schranke angeben:

Theorem 3.11 *Das Problem* LC[0]-FA *ist* NC^1*-hart.*

Dieses Theorem folgt direkt aus Theorem 2.31, da alle Modelle mit genau einer, sich selbst sehenden Welt auch LC[0]-Modelle sind.
Die Logik LC ist ohne Beschränkung der Variablenzahl nicht mehr endlich erzeugt (allein jede atomare Aussage bildet eine eigene Äquivalenzklasse), daher ist anzunehmen, dass das Formelauswertungsproblem auch nicht in NC^1 liegt. Wir zeigen, dass es dennoch unterhalb von P liegt, die exakte Komplexität ist allerdings noch offen.

Theorem 3.12 *Das Problem* LC-FA *ist in* $\mathsf{LOGdetCFL}$.

Beweis. Die wesentliche Idee bei diesem Beweis besteht darin, dass es in linearen Modellen für eine Formel immer eine eindeutig bestimmte Welt gibt, ab der sie erfüllt ist, oder sie ist in keiner Welt erfüllt. Da sich die Modelle nicht verzweigen, muss man für jede Teilformel nur diese eine Welt bestimmen und kann daraus errechnen, ob eine Formel in einer Welt erfüllt ist. Diese Berechnung kann mittels Tiefensuche durch die Formel erfolgen.
Sei $\langle \varphi, \mathcal{M} = (W, \leqslant, \xi, \mathrm{VAR}), w \rangle$ eine LC-FA-Instanz. Zur Vereinfachung gehen wir o.B.d.A. davon aus, dass $W = \{1, 2, \ldots, n\}$ gilt und $\leqslant$ diese Welten in natürlicher Reihenfolge ordnet. Wegen der Monotonieeigenschaft gibt es für jede LC-Formel α ein $i_\alpha \in \{1, 2, \ldots, n, n+1\}$ so, dass α in den Welten $1, 2, \ldots, i_\alpha - 1$ nicht erfüllt und in den Welten $i_\alpha, i_\alpha + 1, \ldots, n$ erfüllt ist. Gilt $i_\alpha = n + 1$, dann ist α in keiner Welt von $\mathcal{M}$ erfüllt. Wir definieren eine Funktion g, die einem Paar bestehend aus einem LC-Modell und einer LC-Formel den entsprechenden i-Wert zuordnet.

$$g(\mathcal{M}, \alpha) \quad := \quad \begin{cases} n+1, & \text{falls} \quad \alpha = \bot \\ \min(\xi(\alpha) \cup \{n+1\}), & \text{falls} \quad \alpha \in \mathrm{VAR} \\ \max(g(\mathcal{M}, \beta), g(\mathcal{M}, \gamma)), & \text{falls} \quad \alpha = \beta \wedge \gamma \\ \min(g(\mathcal{M}, \beta), g(\mathcal{M}, \gamma)), & \text{falls} \quad \alpha = \beta \vee \gamma \\ g(\mathcal{M}, \gamma), & \text{falls} \quad \alpha = \beta \rightarrowtriangle \gamma \;\&\; g(\mathcal{M}, \beta) < g(\mathcal{M}, \gamma) \\ 1, & \text{falls} \quad \alpha = \beta \rightarrowtriangle \gamma \;\&\; g(\mathcal{M}, \beta) \geq g(\mathcal{M}, \gamma) \end{cases}$$

Um $\mathcal{M}, w \models \varphi$ zu entscheiden, müssen wir $g(\mathcal{M}, \varphi)$ berechnen. Die eigentliche Formelauswertung erfolgt mit Algorithmus 4 auf Basis dieser Berechnung. Die Korrektheit folgt direkt aus der Definition von g.

Zum Ressourcenverbrauch von Algorithmus 4 ist zu sagen, dass jede Variable in logarithmischem Platz gespeichert werden kann. Die eingegebene Formel wird rekursiv durchlaufen und dabei gibt es für jede Teilformel höchstens einen rekursiven Aufruf. Die Laufzeit ist also polynomiell. Da die für die Rekursion notwendigen Daten auf einem Stapel gespeichert werden können, kann Algorithmus 4 auf einer Turingmaschine implementiert werden, der die Ressourcen von LOGdetCFL zur Verfügung stehen.

Algorithmus 4 `Formelauswerter für` LC

Eingabe: LC-Formel φ, LC-Modell $\mathcal{M}$, Welt w

1: **wenn** $\texttt{GIndex}(\mathcal{M}, \varphi) \leq w$ **dann** akzeptiere **sonst** lehne ab
2: **Funktion** $\texttt{GIndex}(\mathcal{N} = (\{1, 2, \ldots, n\}, \leqslant, \xi, \mathrm{VAR}), \alpha)$ // berechnet $g(\mathcal{N}, \alpha)$
3: **wenn** $\alpha = \bot$ **dann Rückgabe** $n + 1$
4: **wenn** $\alpha \in \mathrm{VAR}$ **dann Rückgabe** $\min(\xi(\alpha) \cup \{n + 1\})$
5: **wenn** $\alpha = \beta \wedge \gamma$ **dann Rückgabe** $\max(\texttt{GIndex}(\mathcal{N}, \beta), \texttt{GIndex}(\mathcal{N}, \gamma))$
6: **wenn** $\alpha = \beta \vee \gamma$ **dann Rückgabe** $\min(\texttt{GIndex}(\mathcal{N}, \beta), \texttt{GIndex}(\mathcal{N}, \gamma))$
7: **wenn** $\alpha = \beta \twoheadrightarrow \gamma$ **dann**
8: $g := \texttt{GIndex}(\mathcal{N}, \gamma)$
9: **wenn** $\texttt{GIndex}(\mathcal{N}, \beta) < g$ **dann Rückgabe** g **sonst Rückgabe** 1

□

3.4 Zusammenfassung

In Kapitel 3 wurden endlich erzeugte Logiken untersucht und wir konnten zeigen, dass allein die Eigenschaft, dass eine intuitionistische Logik endlich erzeugt ist, hinreichend dafür ist, dass ihr Formelauswertungsproblem in NC^1 liegt (Theorem 3.3). Aus diesem Ergebnis und den verwendeten Beweistechniken konnten wir weitere Resultate ableiten. Die wesentlichen sind, dass auch das Erfüllbarkeits- und das Tautologieproblem für diese Logiken in NC^1 sind (Lemma 3.4).
Für viele Logiken basiert der Beweis dafür, dass sie endlich erzeugt sind, auf der Semantik der Heyting-Algebren. Ist die Grundmenge der Heyting-Algebra endlich, so hat die Logik nur endlich viele Formeläquivalenzklassen und ist damit endlich erzeugt. Wesentliche Ergebnisse dazu liefert [dLHdJ12]. Wir nennen einige dieser Logiken in den Theoremen 3.7 und 3.8 und zeigen, dass ihr Formelauswertungsproblem teilweise NC^1-vollständig und teilweise trivial ist.
Im letzten Abschnitt haben wir uns mit den Fragmenten von LC beschäftigt, in denen nur eine feste Zahl an Variablen vorkommen und konnte zeigen, dass sie alle endlich erzeugt sind (Theorem 3.10). Zusammen mit den Theoremen 3.3 und 3.11 ergibt sich das folgende Resultat.

Theorem 3.13 *Für alle $n \in \mathbb{N}$ ist das Problem* LC[n]-FA NC^1*-vollständig.*

Für LC ohne Einschränkung bleibt die exakte Komplexität des Formelauswertungsproblems offen. Wir zeigen in Theorem 3.12, dass es in $\mathsf{LOGdetCFL}$ liegt. Weitere endlich erzeugte Logiken betrachten wir in Abschnitt 4.4. Dort zeigen wir, dass alle superintuitionistischen Logiken mit einer Variablen endlich erzeugt sind und ein NC^1-vollständiges Formelauswertungsproblem haben.
Dass eine Logik endlich erzeugt ist, ist hinreichend dafür, dass ihr Formelauswertungsproblem in NC^1 liegt. Notwendig ist es hingegen nicht, da selbst die normale Aussagenlogik ein NC^1-vollständiges Formelauswertungsproblem hat, aber nicht endlich erzeugt ist. Trotzdem nimmt sie als (super-)intuitionistische Logik eine Sonderstellung ein, da sie die Kripke-Modelle nur sehr stark eingeschränkt nutzt, hier ist nur eine Welt, die sich selbst sieht, notwendig. Außerdem ist in der Aussagenlogik der intuitionistische Grundgedanke, dass der Satz des ausgeschlossenen Dritten nicht gültig ist, gar nicht mehr verwirklicht. Bei stärkerem Gebrauch der Kripke-Semantik in einer Logik, die nicht endlich erzeugt ist – in der also Modelle beliebiger Größe nötig sein können –, liegt es hingegen schon sehr nahe, dass weder ihr Formelauswertungsproblem, noch ihr Erfüllbarkeits- oder Tautologieproblem in NC^1 liegen.
Die verwendeten Beweistechniken für das Haupttheorem (Theorem 3.3) und auch für die Folgerungen (Lemmata 3.4, 3.5 und 3.6) lassen sich ohne Probleme auf Modallogiken übertragen.

Bemerkung 3.14 *Wenn eine Modallogik* L *endlich erzeugt ist, dann sind die Probleme* L-FA, L-SAT *und* L-TAUT *in* NC^1.

Der Beweis verläuft analog zum Beweis von Theorem 3.3 bzw. zum Beweis von Lemma 3.4. Stellen in den Beweisen, an denen auf die konkrete Konstruktion von Operatoren Bezug genommen wird, können entsprechend dieser Konstruktion immer angepasst werden.

Kapitel 4

IPL mit einer Variablen

In diesem Kapitel befassen wir uns mit der Logik IPL[1] = $(\mathfrak{F}^{\mathsf{i}}[1], \mathfrak{K}^{\mathsf{i}}[1])$. Dabei ist $\mathfrak{K}^{\mathsf{i}}[1] = \{\mathcal{M} = (W, \leqslant, \xi, \{p\}) \mid \mathcal{M} \in \mathfrak{K}^{\mathsf{i}}_{\mathit{refl}}\}$ die Menge aller reflexiven intuitionistischen Modelle, in denen p als einzige Variable vorkommt. Die Formeln enthalten ebenfalls nur p als Variable und die Konstante $\bot$. Das Formelauswertungsproblem für diese Logik ist interessant, weil wir seine AC^1-Vollständigkeit zeigen können. Nach unserer Kenntnis ist es das erste „natürliche“ AC^1-vollständige Problem. Bei bekannten AC^1-vollständigen Problemen (siehe zum Beispiel [BM95]) ist immer eine logarithmische Beschränkung bereits Teil der Problemdefinition (siehe auch Theorem 2.22, hier wird explizit gefordert, dass der Graph nur logarithmisch viele Schichten hat). Interessant ist das Ergebnis außerdem unter dem Gesichtspunkt der Abgrenzung. Wie in Abschnitt 5.1.3 gezeigt wird, ist das Formelauswertungsproblem für IPL[2] bereits P-hart (Bemerkung 5.7). Lässt man hingegen keine Variablen zu, ist es NC^1-vollständig (Theorem 2.31). Schränkt man die Verwendung von $\rightarrowtriangle$ und $\vee$ ein, ist es ebenfalls in NC^1 (siehe dazu Abschnitt 3.3).

Bereits 1960 beschäftigte sich Nishimura [Nis60] mit IPL[1] und zeigte, dass es in dieser Logik unendlich viele Formeläquivalenzklassen gibt. Repräsentanten für die Klassen können leicht induktiv definiert werden (siehe zum Beispiel [Gab81]). Mit Hilfe dieser Repräsentanten kann man auch für die Modelle entsprechende Äquivalenzklassen angeben. Die Formelauswertung erfolgt dann nicht im traditionellen Sinne (Algorithmus 1), sondern man bestimmt zur gegebenen Formel und zum gegebenen Modell den jeweiligen äquivalenten Repräsentanten und überprüft, ob diese zusammenpassen. Algorithmus 6 arbeitet nach diesem Prinzip, berechnet aber nicht mehr die Repräsentanten selbst, sondern nur noch deren Indizes.

Dieses Kapitel ist wie folgt gegliedert: In Abschnitt 4.1 geben wir grundlegende Eigenschaften von IPL[1]-Formeln und -Modellen an. Die obere Schranke von IPL[1]-FA ist Inhalt von Abschnitt 4.2 und in Abschnitt 4.3 bestimmen wir die untere Schranke. Weiter betrachten wir in Abschnitt 4.4 superintuitionistische Logiken mit einer Variablen. In Abschnitt 4.5 untersuchen wir einige Varianten von IPL[1] und Abschnitt 4.6 ist eine Zusammenfassung dieses Kapitels.

Wenn es nicht anders gesagt wird, meint der Begriff „äquivalent“ in Bezug auf Formeln in diesem Kapitel immer $\mathfrak{K}^{\mathsf{i}}[1]$-äquivalent. Ebenso meint „äquivalent“ in Bezug auf Welten immer $\mathfrak{F}^{\mathsf{i}}[1]$-äquivalent.

4.1 Grundlegende Eigenschaften

In diesem Abschnitt werden wir einige grundlegende Eigenschaften der IPL[1]-Formeln und -Modelle angeben. Diese basieren im Wesentlichen auf einer Arbeit von Nishimura [Nis60].

4.1.1 Die Formeln aus $\mathfrak{F}^{\mathsf{i}}[1]$

Die Menge $\mathfrak{F}^{\mathsf{i}}[1]$ aller IPL[1]-Formeln lässt sich in unendlich viele Äquivalenzklassen zerlegen [Nis60]. Repräsentanten dieser Klassen können induktiv definiert werden (siehe zum Beispiel [Gab81]).

Definition 4.1 *Es seien*

$$\varphi_1 := \neg\neg p \ , \qquad \psi_1 := p \ ,$$
$$\varphi_{n+1} := \varphi_n \rightarrowtriangle \psi_n \quad \textit{und} \quad \psi_{n+1} := \varphi_n \vee \psi_n \qquad \textit{für } n \geq 1 \ .$$

Die Formeln $\bot, \top, \varphi_1, \psi_1, \varphi_2, \psi_2, \ldots$ *heißen Rieger-Nishimura-Formeln.*

Wir geben Rieger-Nishimura-Formeln im Weiteren immer mit den fett gedruckten griechischen Buchstaben $\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\varphi}$ und $\boldsymbol{\psi}$ an.

Theorem 4.2 ([Nis60],[Gab81, Kap. 6.1,Th. 7]) *Jede* IPL[1]*-Formel ist zu genau einer Rieger-Nishimura-Formel äquivalent. Die Rieger-Nishimura-Formeln sind alle paarweise nicht äquivalent.*

Die Äquivalenzklassen von $\mathfrak{F}^{\mathsf{i}}[1]$ bezüglich $\equiv_{\mathfrak{K}^{\mathsf{i}}[1]}$ bilden die freie Heyting-Algebra $(\{p\}, \sqcap, \sqcup, \multimap, \bot)$ über dem Generator p (siehe [MT46, Nis60]). Dabei sind $\sqcap$, $\sqcup$ und $\multimap$ wie in Abschnitt 2.1.3 definiert. Statt der Menge aller Elemente wird als erstes Element der Algebra nur die Menge der Generatoren $\{p\}$ angegeben. Die Elemente sind die Äquivalenzklassen aller Formeln, die aus p und $\bot$ und den Operatoren gebildet werden können. Die Elemente bilden die Äquivalenzklassenzerlegung von $\mathfrak{F}^{\mathsf{i}}[1]$ bezüglich $\equiv_{\mathfrak{K}^{\mathsf{i}}[1]}$. Diese Heyting-Algebra heißt auch Rieger-Nishimura-Verband. Abbildung 4.1 gibt eine graphische Vorstellung von diesem Verband und der induzierten Halbordnung.
Nishimura [Nis60] zeigte für die Operationen, dass $[\alpha] \sqcap [\beta] = [\alpha \wedge \beta]$, $[\alpha] \sqcup [\beta] = [\alpha \vee \beta]$ und $[\alpha] \multimap [\beta] = [\alpha \rightarrowtriangle \beta]$ gilt. Daraus kann man für die Rieger-Nishimura-Formeln das folgende Theorem leicht ableiten (siehe auch[Gab81]).

Theorem 4.3 ([Nis60, Gab81]) *Sei* $\boldsymbol{\alpha}$ *eine beliebige Rieger-Nishimura-Formel, dann gelten folgende Äquivalenzen:*

$$\varphi_n \rightarrowtriangle \varphi_n \equiv_{\mathfrak{K}^{\mathsf{i}}[1]} \top \ , \qquad \varphi_n \vee \varphi_n \equiv_{\mathfrak{K}^{\mathsf{i}}[1]} \varphi_n \ ,$$
$$\varphi_n \rightarrowtriangle \varphi_{n+1} \equiv_{\mathfrak{K}^{\mathsf{i}}[1]} \varphi_{n+1} \ , \qquad \varphi_n \vee \varphi_{n+1} \equiv_{\mathfrak{K}^{\mathsf{i}}[1]} \psi_{n+2} \ ,$$
$$\varphi_n \rightarrowtriangle \varphi_{n+k} \equiv_{\mathfrak{K}^{\mathsf{i}}[1]} \top \textit{ für } k > 1 \ , \qquad \varphi_n \vee \varphi_{n+k} \equiv_{\mathfrak{K}^{\mathsf{i}}[1]} \varphi_{n+k} \textit{ für } k > 1 \ ,$$

$\varphi_{n+k} \rightarrowtriangle \varphi_n \equiv_{\mathfrak{K}^i[1]} \varphi_n$ *für* $k \geq 1$,
$\varphi_n \rightarrowtriangle \psi_n \equiv_{\mathfrak{K}^i[1]} \varphi_{n+1}$,
$\varphi_n \rightarrowtriangle \psi_{n+k} \equiv_{\mathfrak{K}^i[1]} \top$ *für* $k \geq 1$,
$\varphi_{n+1} \rightarrowtriangle \psi_n \equiv_{\mathfrak{K}^i[1]} \varphi_{n+2}$,
$\varphi_{n+2} \rightarrowtriangle \psi_n \equiv_{\mathfrak{K}^i[1]} \varphi_{n+1}$,
$\varphi_{n+k} \rightarrowtriangle \psi_n \equiv_{\mathfrak{K}^i[1]} \psi_n$ *für* $k > 2$,
$\psi_n \rightarrowtriangle \psi_{n+k} \equiv_{\mathfrak{K}^i[1]} \top$ *für* $k \geq 0$,
$\psi_{n+1} \rightarrowtriangle \psi_n \equiv_{\mathfrak{K}^i[1]} \varphi_{n+1}$,
$\psi_{n+k} \rightarrowtriangle \psi_n \equiv_{\mathfrak{K}^i[1]} \psi_n$ *für* $k > 1$,
$\psi_{n+k} \rightarrowtriangle \varphi_n \equiv_{\mathfrak{K}^i[1]} \varphi_n$ *für* $k \geq 0$,
$\psi_n \rightarrowtriangle \varphi_{n+k} \equiv_{\mathfrak{K}^i[1]} \top$ *für* $k \geq 1$,
$\varphi_1 \rightarrowtriangle \bot \equiv_{\mathfrak{K}^i[1]} \varphi_2$,
$\varphi_2 \rightarrowtriangle \bot \equiv_{\mathfrak{K}^i[1]} \varphi_1$,
$\varphi_n \rightarrowtriangle \bot \equiv_{\mathfrak{K}^i[1]} \bot$ *für* $n > 2$,
$\psi_1 \rightarrowtriangle \bot \equiv_{\mathfrak{K}^i[1]} \varphi_1$,
$\psi_n \rightarrowtriangle \bot \equiv_{\mathfrak{K}^i[1]} \bot$ *für* $n > 1$,
$\alpha \rightarrowtriangle \top \equiv_{\mathfrak{K}^i[1]} \top$,
$\top \rightarrowtriangle \alpha \equiv_{\mathfrak{K}^i[1]} \alpha$,
$\bot \rightarrowtriangle \alpha \equiv_{\mathfrak{K}^i[1]} \top$,

$\varphi_n \vee \psi_n \equiv_{\mathfrak{K}^i[1]} \psi_{n+1}$,
$\varphi_n \vee \psi_{n+k} \equiv_{\mathfrak{K}^i[1]} \psi_{n+k}$ *für* $k \geq 1$,
$\varphi_{n+k} \vee \psi_n \equiv_{\mathfrak{K}^i[1]} \varphi_{n+k}$ *für* $k \geq 1$,
$\psi_n \vee \psi_m \equiv_{\mathfrak{K}^i[1]} \psi_{\max\{n,m\}}$,
$\alpha \vee \bot \equiv_{\mathfrak{K}^i[1]} \alpha$,
$\alpha \vee \top \equiv_{\mathfrak{K}^i[1]} \top$,

$\varphi_n \wedge \varphi_n \equiv_{\mathfrak{K}^i[1]} \varphi_n$,
$\varphi_1 \wedge \varphi_2 \equiv_{\mathfrak{K}^i[1]} \bot$,
$\varphi_n \wedge \varphi_{n+1} \equiv_{\mathfrak{K}^i[1]} \psi_{n-1}$ *für* $n > 1$,
$\varphi_n \wedge \varphi_{n+k} \equiv_{\mathfrak{K}^i[1]} \varphi_n$ *für* $k > 1$,
$\varphi_1 \wedge \psi_1 \equiv_{\mathfrak{K}^i[1]} \bot$,
$\varphi_n \wedge \psi_n \equiv_{\mathfrak{K}^i[1]} \psi_{n-1}$ *für* $n > 1$,
$\varphi_n \wedge \psi_{n+k} \equiv_{\mathfrak{K}^i[1]} \varphi_n$ *für* $k \geq 1$,
$\varphi_{n+k} \wedge \psi_n \equiv_{\mathfrak{K}^i[1]} \psi_n$ *für* $k \geq 1$,
$\psi_n \wedge \psi_m \equiv_{\mathfrak{K}^i[1]} \psi_{\min\{n,m\}}$,
$\alpha \wedge \bot \equiv_{\mathfrak{K}^i[1]} \bot$, *und*
$\alpha \wedge \top \equiv_{\mathfrak{K}^i[1]} \alpha$.

(Durch die Kommutativität von $\vee$ *und* $\wedge$ *ergeben sich die restlichen Fälle.)*

Ein Beweis ist in [Wei08] zu finden. Mit Hilfe der Theoreme 4.2 und 4.3 treffen wir später eine Aussage über die Länge von IPL[1]-Formeln. Definition 4.1 zeigt bereits, dass sie sich bei Rieger-Nishimura-Formeln von „Stufe zu Stufe“ verdoppelt. Wir übertragen dieses exponentielle Wachstum in Lemma 4.7 auf alle Formeln.

4.1.2 Die Modelle aus $\mathfrak{K}^i[1]$

Die Menge $\mathfrak{K}^i[1]$ ist die Menge aller IPL[1]-Modelle. Ähnlich wie bei den IPL[1]-Formeln, die sich durch äquivalente Rieger-Nishimura-Formeln repräsentieren lassen, kann man für die IPL[1]-Modelle kanonische Modelle definieren. Diese Modelle werden so konstruiert, dass jede Welt eines beliebigen IPL[1]-Modells zur untersten Welt von genau einem dieser kanonischen Modelle äquivalent ist. Der Zusammenhang zwischen den Rieger-Nishimura-Formeln und den kanonischen Modellen wird in Lemma 4.5 beschrieben. Wir definieren diese Modelle wie folgt.

Definition 4.4 *Für* $n \geq 1$ *sei* $\mathcal{H}_n := (W_n, \trianglelefteq, \xi_n, \{p\})$ *mit*

(1) $W_n := \{1, 2, \ldots, n-2\} \cup \{n\}$,

(2) $\trianglelefteq := \{(w, v) \mid w, v \in W_n,\ w = v$ *oder* $w \geq v + 2\}$ *und*

(3) $\xi_n(p) := \begin{cases} \varnothing, & \text{falls} \quad n = 2 \\ \{1\}, & \text{sonst} . \end{cases}$

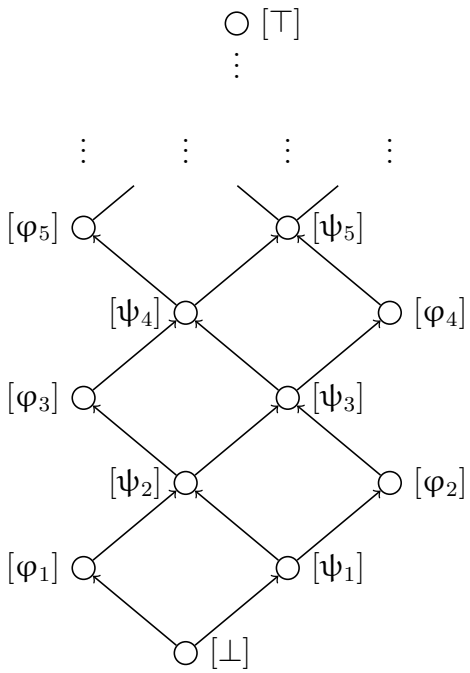

Abbildung 4.1: Der Rieger-Nishimura-Verband. Zum Beispiel gilt $[\psi_3] \multimap [\varphi_3] = [\varphi_4]$, weil $[\varphi_4]$ das größte Element z ist, mit $[\varphi_3] \sqcap z \sqsubseteq [\psi_3]$ (es gilt $[\varphi_3] \sqcap [\varphi_4] = [\psi_2] \sqsubseteq [\psi_3]$). In der Sprache der IPL[1]-Formeln ausgedrückt, bedeutet dies $\psi_3 \rightarrowtriangle \varphi_3 \equiv_{\mathfrak{R}^i[1]} \varphi_4$.

Wir bezeichnen die Modelle $\mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \ldots$ *als kanonische Modelle.*

Abbildung 4.2 zeigt die Modelle $\mathcal{H}_9$ und $\mathcal{H}_{10}$. Für die kanonischen Modelle und die Rieger-Nishimura-Formeln gilt der folgende Zusammenhang.

Lemma 4.5 *Für* $n \geq 1$ *und* $k \geq 1$ *gilt*

(1) $\mathcal{H}_n, n \models \psi_k \iff n \leq k$ *und*

(2) $\mathcal{H}_n, n \models \varphi_k \iff n < k$ *oder* $n = k + 1$.

Beweis. Wir verwenden folgende wesentliche Eigenschaft der kanonischen Modelle: Seien $x \geq 1$, $y \geq 1$ und $z \in W_x \cap W_y$, dann gilt $(\mathcal{H}_x, z) \equiv_{\mathfrak{F}^i[1]} (\mathcal{H}_y, z)$. Dies folgt direkt aus der Konstruktion der kanonischen Modelle. Anschaulich bedeutet dies, dass $\mathcal{H}_n$ alle kanonischen Modelle $\mathcal{H}_i$ mit $i \leq n - 2$ enthält.
Den Beweis von (1) und (2) führen wir mit vollständiger Induktion über n. Für $n \in \{1, 2, 3\}$ folgen beide Aussagen direkt aus der Konstruktion der kanonischen Modelle und der Rieger-Nishimura-Formeln.

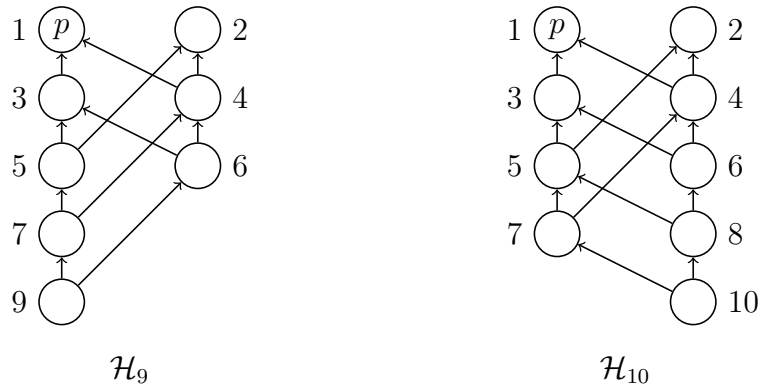

Abbildung 4.2: Die kanonischen Modelle $\mathcal{H}_9$ (links) und $\mathcal{H}_{10}$ (rechts). Transitive und reflexive Kanten sind nicht abgebildet.

Für den Induktionsschritt sei $n > 4$. Nach Induktionsvoraussetzung gilt für alle $m < n$

$$\mathcal{H}_m, m \models \psi_m, \psi_{m+1}, \dots \ , \qquad \text{(i)}$$

$$\mathcal{H}_m, m \not\models \psi_1, \psi_2, \dots, \psi_{m-1} \ , \qquad \text{(ii)}$$

$$\mathcal{H}_m, m \models \varphi_{m-1}, \varphi_{m+1}, \varphi_{m+2}, \dots \ , \qquad \text{(iii)}$$

$$\mathcal{H}_m, m \not\models \varphi_1, \varphi_2, \dots, \varphi_{m-2}, \varphi_m \ . \qquad \text{(iv)}$$

Nach Konstruktion von $\mathcal{H}_n$ gilt $n-2 \trianglelefteq n$, $n-3 \trianglelefteq n$ und $n-1 \not\trianglelefteq n$. Damit und mit der Induktionsvoraussetzung können wir auf

$$\mathcal{H}_n, n \not\models \psi_1, \psi_2, \dots, \psi_{n-3} \ , \qquad \text{(v)}$$

$$\mathcal{H}_n, n \not\models \varphi_1, \varphi_2, \dots, \varphi_{n-5}, \varphi_{n-3} \ , \qquad \text{(vi)}$$

$$\mathcal{H}_n, n \not\models \varphi_1, \varphi_2, \dots, \varphi_{n-4}, \varphi_{n-2} \qquad \text{(vii)}$$

schließen. Aus $\mathcal{H}_n, n \not\models \psi_{n-3}$ (v) und $\mathcal{H}_n, n \not\models \varphi_{n-3}$ (vi) folgt wegen $\psi_{n-2} = \varphi_{n-3} \vee \psi_{n-3}$

$$\mathcal{H}_n, n \not\models \psi_{n-2} \ . \qquad \text{(viii)}$$

Daraus und aus $\mathcal{H}_n, n \not\models \varphi_{n-2}$ (vii) folgt mit $\psi_{n-1} = \varphi_{n-2} \vee \psi_{n-2}$

$$\mathcal{H}_n, n \not\models \psi_{n-1} \ . \qquad \text{(ix)}$$

Aus (v), (viii) und (ix) folgt die Richtung „$\Rightarrow$“ von (1).
Aus (i) folgt $\mathcal{H}_n, i \models \psi_{n-2}$ für $i \leq n-2$ und aus (vii) folgt $\mathcal{H}_n, n \not\models \varphi_{n-2}$. Damit gilt wegen $\varphi_{n-1} = \varphi_{n-2} \twoheadrightarrow \psi_{n-2}$

$\mathcal{H}_n, n \models \varphi_{n-1}$. (x)

Daraus und aus $\mathcal{H}_n, n \not\models \psi_{n-1}$ (ix), folgt mit $\varphi_n = \varphi_{n-1} \rightarrowtriangle \psi_{n-1}$

$\mathcal{H}_n, n \not\models \varphi_n$. (xi)

Mit (vi), (vii) und (xi) haben wir die Richtung „⇒“ von (2) gezeigt.
Für „⇐“ von (1) sei $i \geq 0$. Dann gelten folgende Äquivalenzen:

$$
\begin{aligned}
&\mathcal{H}_n, n \models \psi_{n+i} \\
\Leftrightarrow\quad &\mathcal{H}_n, n \models \varphi_{n+i-1} \vee \psi_{n+i-1} \\
\Leftrightarrow\quad &\mathcal{H}_n, n \models \varphi_{n+i-1} \vee (\varphi_{n+i-2} \vee \psi_{n+i-2}) \\
&\vdots \\
\Leftrightarrow\quad &\mathcal{H}_n, n \models \varphi_{n+i-1} \vee (\varphi_{n+i-2} \vee (\cdots \vee (\varphi_{n-1} \vee \psi_{n-1}) \ldots)) \\
\Leftrightarrow\quad &\mathcal{H}_n, n \models \varphi_{n+i-1} \vee \varphi_{n+i-2} \vee \cdots \vee \varphi_{n-1} \vee \psi_{n-1}
\end{aligned}
$$

Nach (x) gilt $\mathcal{H}_n, n \models \varphi_{n-1}$ und es folgt

$\mathcal{H}_n, n \models \psi_{n+i}$, (xii)

womit die Richtung „⇐“ von (1) gezeigt ist.
Für „⇐“ von (2) sei $j \geq 1$. Es gilt $\varphi_{n+j} = \varphi_{n+j-1} \rightarrowtriangle \psi_{n+j-1}$. Nach (i) und (xi) ist ψ_{n+j-1} in allen Welten von $\mathcal{H}_n$ erfüllt und es folgt

$\mathcal{H}_n, n \models \varphi_{n+j}$. (xiii)

Die Richtung „⇐“ von (2) folgt aus (x) und (xiii). □

Eine ähnliche Version von Lemma 4.5 gibt Gabbay 1981 in Kapitel 6.1 Lemma 10 von [Gab81] an, für eine übersichtliche Darstellung siehe Tabelle 4.2. Wir verallgemeinern dieses Lemma in Bemerkung 4.14 mit Hilfe von Lemma 4.13 für beliebige Formeln und Modelle.

4.2 Obere Schranken

Wir zeigen in diesem Abschnitt, dass das Formelauswertungsproblem für IPL[1] in AC^1 ist. Jede IPL[1]-Formel ist zu genau einer Rieger-Nishimura-Formel äquivalent und kann deswegen durch den Index und den Typ (φ, ψ oder Konstante) repräsentiert werden. Das Paar aus Index und Typ nennen wir im Folgenden *Rieger-Nishimura-Index*. Wir zeigen, dass die Größe des Indexes einer Formel höchstens logarithmisch in der Länge der Formel ist. Weiter zeigen wir, dass die Frage, ob ein gegebener Index zu einer Formel passt, mit den Ressourcen von $\mathsf{LOGdetCFL}$ beantwortet werden kann. Im zweiten Teil schauen wir uns die Modelle an. Wir definieren eine Funktion $\hbar$, die eine Welt w eines Modells $\mathcal{M}$ genau dann auf die

$\models$	$\bot$	φ_1	ψ_1	φ_2	ψ_2	$\cdots$	φ_{n-1}	ψ_{n-1}	φ_n	ψ_n	$\cdots$
$\mathcal{H}_1$	×	×	✓	✓	✓	✓	✓	✓	✓	✓	✓
$\mathcal{H}_2$	×	✓	×	×	✓	✓	✓	✓	✓	✓	✓
$\mathcal{H}_3$	×	×	×	✓	×	$\cdots$	✓	✓	✓	✓	✓
$\vdots$	×	×	×	×	×	$\ddots$	✓	✓	✓	✓	✓
$\mathcal{H}_{n-2}$	×	×	×	×	×	$\cdots$	✓	✓	✓	✓	✓
$\mathcal{H}_{n-1}$	×	×	×	×	×	$\cdots$	×	✓	✓	✓	✓
$\mathcal{H}_n$	×	×	×	×	×	×	✓	×	×	✓	✓
$\mathcal{H}_{n+1}$	×	×	×	×	×	×	×	×	✓	×	$\cdots$
$\vdots$	×	×	×	×	×	×	×	×	×	×	$\cdots$

Tabelle 4.2: Hier wird Lemma 4.5 veranschaulicht. Zum Beispiel wird φ_2 in $\mathcal{H}_3, 3$ erfüllt, während $\mathcal{H}_3, 3 \not\models \psi_3$ gilt.

positive natürliche Zahl i abbildet, wenn $(\mathcal{M}, w) \equiv_{\mathfrak{F}^i[1]} (\mathcal{H}_i, i)$ gilt. Die Zahl i heißt in diesem Fall *Modellindex* von w in $\mathcal{M}$. Die Entscheidung, ob eine Zahl der Modellindex einer Welt ist, kann mit einer alternierenden Turingmaschine in logarithmischem Platz gefällt werden. Bemerkung 4.14 verallgemeinert Lemma 4.5 und liefert einen Zusammenhang zwischen den Rieger-Nishimura-Indizes und den Modellindizes. Diesen Zusammenhang nutzen wir im letzten Unterabschnitt, um einen Algorithmus für das Formelauswertungsproblem anzugeben.

4.2.1 Der Rieger-Nishimura-Index

Wir definieren die Funktion *RNIndex*, die eine IPL[1]-Formel auf das Paar bestehend aus Index und Typ der äquivalenten Rieger-Nishimura-Formel abbildet.

Definition 4.6 *Die Funktion RNIndex* : $\mathfrak{F}^i[1] \mapsto \mathbb{N} \times \{\bot, \top, phi, psi\}$ *wird wie folgt definiert: Sei* $\alpha \in \mathfrak{F}^i[1]$*, dann ist*

$$RNIndex(\alpha) \quad := \quad \begin{cases} (i, phi), & falls \quad \alpha \equiv_{\mathfrak{K}^i[1]} \varphi_i \\ (i, psi), & falls \quad \alpha \equiv_{\mathfrak{K}^i[1]} \psi_i \\ (0, \bot), & falls \quad \alpha \equiv_{\mathfrak{K}^i[1]} \bot \\ (0, \top), & falls \quad \alpha \equiv_{\mathfrak{K}^i[1]} \top \ . \end{cases}$$

$RNIndex(\alpha)$ ist der Rieger-Nishimura-Index und $Rang(\alpha) = i$ der Rang von α, falls $RNIndex(\alpha) = (i, x)$ gilt.

Wir geben jetzt eine untere Schranke für der Länge von Formeln abhängig von ihrem Rang an.

Lemma 4.7 *Sei* $\alpha \in \mathfrak{F}^{\mathrm{i}}[1]$. *Dann gilt* $Rang(\alpha) \leq c \cdot \log(|\alpha|)$ *für eine Konstante* c *unabhängig von* α.

Beweis. Der Beweis basiert im Wesentlichen auf den in Theorem 4.3 angegebenen Äquivalenzen. Für eine Äquivalenz $\alpha \star \beta \equiv_{\mathfrak{K}^{\mathrm{i}}[1]} \gamma$ aus diesem Theorem gilt, dass der Rang von γ höchstens um eins größer ist als das Maximum der Ränge von β und γ. Und wenn γ einen echt größeren Rang als α und β hat, dann unterscheiden sich deren Ränge ebenfalls um höchstens eins. Damit ergibt sich die folgende Behauptung.

Behauptung 4.1 *Sei* $\alpha \in \mathfrak{F}^{\mathrm{i}}[1]$. *Dann gilt* $|\alpha| \geq fib(Rang(\alpha))$.[1]

Beweis der Behauptung. Wir zeigen diese Behauptung mit vollständiger Induktion über die Länge von α. Den Induktionsanfang bilden die Formeln der Länge 1. Für $\alpha \in \{\bot, \top, p\}$ folgt die Behauptung aus $fib(0) = fib(1) = 1$.
Für den Induktionsschritt sei $\alpha = \beta \star \gamma$ mit $\star \in \{\wedge, \vee, \rightarrowtriangle\}$ und $|\alpha| > 1$. Es gilt $|\alpha| = |\beta| + |\gamma| + 1$ und nach Induktionsvoraussetzung gilt dann $|\alpha| \geq fib(Rang(\beta)) + fib(Rang(\gamma)) + 1$. Wir unterscheiden jetzt die folgende Fälle:

(i) $\alpha \in [\bot] \cup [\top]$. In diesem Fall gilt $Rang(\alpha) = 0$ und die Behauptung folgt sofort.

(ii) $\gamma \in [\bot]$. Wir müssen hier nur noch die Fälle $\star = \vee$ und $\star = \rightarrowtriangle$ unterscheiden:

Falls $\star = \vee$ ist, folgt sofort $\beta \equiv_{\mathfrak{K}^{\mathrm{i}}[1]} \alpha$ und damit auch $Rang(\beta) = Rang(\alpha)$. Nach Induktionsvoraussetzung gilt $|\alpha| \geq fib(Rang(\beta)) = fib(Rang(\alpha))$.

Falls $\star = \rightarrowtriangle$ ist, dann gilt $\beta \in [\varphi_1] \cup [\varphi_2] \cup [\psi_1]$ und $\alpha \in [\varphi_2] \cup [\varphi_1]$ (siehe dazu Theorem 4.3). Also folgt $|\alpha| \geq |\beta| + 2 > 2 = fib(2) \geq fib(Rang(\alpha))$.

(iii) $\beta \in [\bot]$. Hier kann nur $\star = \vee$ gelten und dieser Fall ist analog zu $\gamma \in [\bot]$.

(iv) $\beta \in [\top]$. In diesem Fall gilt $\star \in \{\rightarrowtriangle, \wedge\}$. In beiden Fällen ist $\alpha \equiv_{\mathfrak{K}^{\mathrm{i}}[1]} \beta$ und mit der Induktionsvoraussetzung folgt $|\alpha| \geq fib(Rang(\alpha))$ sofort.

(v) $\gamma \in [\top]$. Hier gilt $\star = \wedge$ und damit auch $\alpha \equiv_{\mathfrak{K}^{\mathrm{i}}[1]} \gamma$. Die Behauptung folgt wieder aus der Induktionsvoraussetzung.

(vi) Die anderen Fälle. Nach Induktionsvoraussetzung gilt $|\alpha| \geq fib(Rang(\beta)) + fib(Rang(\gamma))$. Bei der Betrachtung aller möglichen Äquivalenzen aus Theorem 4.3, sieht man, dass noch 2 Fälle unterschieden werden müssen:

(a) $Rang(\alpha) \leq Rang(\beta)$ oder $Rang(\alpha) \leq Rang(\gamma)$. Dann gilt nach Induktionsvoraussetzung $|\alpha| > |\beta| \geq fib(Rang(\beta)) \geq fib(Rang(\alpha))$.

[1] $fib(n)$ bezeichnet die n-te Fibonaccizahl, es gilt $fib(0) = 1$, $fib(1) = 1$ und $fib(n+2) = fib(n+1) + fib(n)$ für $n \in \mathbb{N}$.

(b) $Rang(\alpha) > Rang(\beta)$ und $Rang(\alpha) > Rang(\gamma)$. In diesem Fall ist der Rang von höchstens einer der Formeln β und γ gleich $Rang(\alpha) - 2$ und der Rang der anderen Formel ist $Rang(\alpha) - 1$. (Zum Beispiel $\varphi_{k-1} \rightarrowtriangle \psi_{k-2} \equiv_{\mathfrak{K}^{\mathsf{i}}[1]} \varphi_k$ für $k \geq 2$.) Deswegen gilt nach Induktionsvoraussetzung $|\alpha| \geq fib(Rang(\alpha) - 2) + fib(Rang(\alpha) - 1) = fib(Rang(\alpha))$.

Damit ist die Behauptung für alle $\alpha \in \mathfrak{F}^{\mathsf{i}}[1]$ gezeigt. ▪
Behauptung 4.1 zeigt $|\alpha| \geq fib(Rang(\alpha))$. Da die Fibonaccizahlen exponentiell wachsen ($fib(n) \geq \Phi^n$, wobei $\Phi = 1.618\ldots$ den Goldenen Schnitt bezeichnet), folgt $Rang(\alpha) \leq c \cdot \log(|\alpha|)$, wobei c unabhängig von α ist. □

Wir wollen nun den Rieger-Nishimura-Index unter komplexitätstheoretischen Gesichtspunkten betrachten. Dafür definieren wir folgendes Entscheidungsproblem:

Problem:	ÄQRN-FORMEL
Eingabe:	$\langle \alpha, (i, x) \rangle$, wobei $\alpha \in \mathfrak{F}^{\mathsf{i}}[1]$ eine Formel und $(i, x) \in (\mathbb{N} \times \{\bot, \top, phi, psi\})$ ein Rieger-Nishimura-Index ist.
Frage:	Gilt $Rang(\alpha) = (i, x)$?

Lemma 4.8 *Das Problem* ÄQRN-FORMEL *ist in* LOGdetCFL.

Beweis. Wir geben Algorithmus 5 basierend auf den Äquivalenzen aus Theorem 4.3 an. Genauso wie die Verbandsoperationen für Äquivalenzklassen von $\mathfrak{F}^{\mathsf{i}}[1]$ definiert sind, kann man sie auch für die Rieger-Nishimura-Indizes definieren. Seien $\alpha, \beta, \gamma \in \mathfrak{F}^{\mathsf{i}}[1]$ und $\star \in \{\sqcap, \sqcup, \multimap\}$, dann ist $RNIndex(\alpha) \star RNIndex(\beta) = RNIndex(\gamma)$, wenn $[\alpha] \star [\beta] = [\gamma]$ gilt. Damit und aus Theorem 4.3 folgt die Korrektheit des Algorithmus.
Aus Lemma 4.7 folgt, dass jede Variable, die in Algorithmus 5 vorkommt, in logarithmischem Platz gespeichert werden kann. (Genau genommen folgt sogar, dass loglog Speicherplatz ausreicht.) Der Algorithmus durchläuft die eingegebene Formel α rekursiv und berechnet den Rieger-Nishimura-Index jeder Teilformel genau einmal. Dafür benötigt er polynomielle Laufzeit, da die Verbandsoperationen in $\log \log$ Speicherplatz mit einem Blick in Fallunterscheidung in Theorem 4.3 berechnet werden können. Die für die Rekursion nötigen Informationen können auf einem Stapel gespeichert werden. Demnach ist es möglich, Algorithmus 5 auf einer Turingmaschine zu implementieren, der die Ressourcen von LOGdetCFL zur Verfügung stehen.

□

Mit Hilfe von Lemma 4.8 lässt sich auch eine Aussage für die obere Schranke des Tautologieproblems für IPL[1] machen. Eine IPL[1]-Formel ist genau dann eine Tautologie, wenn sie den Rieger-Nishimura-Index $(0, \top)$ hat.

Bemerkung 4.9 *Das Problem* IPL[1]-TAUT *ist in* LOGdetCFL.

Algorithmus 5 `Rieger-Nishimura-Index-Tester`

Eingabe: Formel $\alpha \in \mathfrak{F}^{\mathsf{i}}[1]$, Rieger-Nishimura-Index (i, x)

1: **wenn** $\texttt{RNIndex}(\alpha) = (i, x)$ **dann** akzeptiere **sonst** lehne ab

2: **Funktion** $\texttt{RNIndex}(\beta)$ // liefert Rieger-Nishimura-Index zurück
3: **wenn** $\beta = p$ **dann Rückgabe** $(1, psi)$
4: **wenn** $\beta = \top$ **dann Rückgabe** $(0, \top)$
5: **wenn** $\beta = \bot$ **dann Rückgabe** $(0, \bot)$
6: **wenn** $\beta = \gamma \wedge \delta$ **dann Rückgabe** $\texttt{RNIndex}(\gamma) \sqcap \texttt{RNIndex}(\delta)$
7: **wenn** $\beta = \gamma \vee \delta$ **dann Rückgabe** $\texttt{RNIndex}(\gamma) \sqcup \texttt{RNIndex}(\delta)$
8: **wenn** $\beta = \gamma \multimap \delta$ **dann Rückgabe** $\texttt{RNIndex}(\gamma) \multimap \texttt{RNIndex}(\delta)$

Die Menge der erfüllbaren IPL[1]-Formeln ist gleich der Menge der aussagenlogisch erfüllbaren Formeln mit nur einer Variablen. Da aber in der Aussagenlogik nur zwei mögliche Belegungen getestet werden müssen, kann IPL[1]-SAT auf BFVP$[\wedge, \vee, \rightarrow]$ reduziert werden.

Bemerkung 4.10 *Das Problem* IPL[1]-SAT *ist in* NC^1.

Eine IPL[1]-Formel ist genau dann erfüllbar, wenn sie nicht äquivalent zu $\bot$ ist. Damit lässt sich das Erfüllbarkeitsproblem auch lösen, indem man den Rieger-Nishimura-Index einer Formel bestimmt. Betrachtet man die Formeläquivalenzen der Rieger-Nishimura-Formeln aus Theorem 4.3, ist es nicht ersichtlich, warum der Test, ob eine Formel äquivalent zu $\top$ ist, schwieriger sein sollte, als die Frage, ob sie äquivalent zu $\bot$ ist. Diese Überlegung legt nahe, dass auch IPL[1]-TAUT in NC^1 liegt, allerdings ist ein Beweis für diese Vermutung ist noch offen.

4.2.2 Der Modellindex

In diesem Abschnitt definieren wir die Funktion $\hbar$, die den Welten der IPL[1]-Modelle einen Index zuordnet. Was der *RNIndex* für die IPL[1]-Formeln ist, ist $\hbar$ für die IPL[1]-Modelle.

Definition 4.11 *Es seien* $\mathcal{M} = (W, \leqslant, \xi, \{p\}) \in \mathfrak{K}^{\mathsf{i}}[1]$ *und* w *eine Welt aus* $\mathcal{M}$. *Wir definieren die Funktion* $\hbar : \{(\mathcal{M}, w) \mid \mathcal{M} \in \mathfrak{K}^{\mathsf{i}}[1],\ w \text{ ist Welt aus } \mathcal{M}\} \mapsto \mathbb{N}^+$ *wie folgt:*

$$\hbar(\mathcal{M}, w) \quad := \quad \begin{cases} 1, & \textit{falls} \quad w \in \xi(p) \\ 2, & \textit{falls} \quad w \notin \xi(p) \ \textit{und} \ \forall v \in W_{w\uparrow} : v \notin \xi(p) \\ 3, & \textit{falls} \quad w \notin \xi(p) \ \textit{und} \ \forall v \in W_{w\uparrow} : \hbar(\mathcal{M}, v) \neq 2 \\ & \qquad\qquad \textit{und} \ \exists u \in W_{w\uparrow} : \hbar(\mathcal{M}, u) = 1 \\ n+2, & \textit{falls} \quad \forall v \in W_{w\uparrow} : \quad \hbar(\mathcal{M}, v) \neq n+1 \ \textit{und} \\ & \qquad \exists u_1, u_2 \in W_{w\uparrow} : \hbar(\mathcal{M}, u_1) = n \ \textit{und} \\ & \qquad\qquad\qquad\qquad \hbar(\mathcal{M}, u_2) = n - 1 \ . \end{cases}$$

Falls $\hbar(\mathcal{M}, w) = i$ gilt, bezeichnen wir i als Modellindex von w in $\mathcal{M}$.

Da $\{\hbar(\mathcal{M}, v) \mid v \in W_{w\Uparrow}\} = \{1, 2, \ldots, \hbar(\mathcal{M}, w) - 2\} \cup \{\hbar(\mathcal{M}, w)\}$ gilt, ist $\hbar$ eine korrekt definierte Funktion. Ein Beispiel für die Berechnung von $\hbar$ ist in Abbildung 4.3 zu sehen. Aus der Konstruktion der Funktion $\hbar$ folgt für die kanonischen Modelle sofort das folgende Lemma.

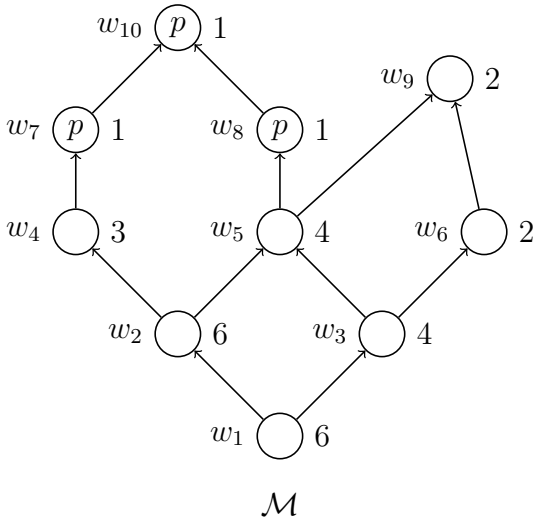

Abbildung 4.3: Ein IPL[1]-Modell $\mathcal{M}$ mit den Welten w_1 bis w_{10} und den Modellindizes rechts von jeder Welt. Zum Beispiel gilt $\hbar(\mathcal{M}, w_1) = 6$, weil w_1 einen Nachfolger mit Modellindex 3 (die Welt w_4), einen mit Modellindex 4 (die Welten w_3 und w_5) und keinen Nachfolger mit Modellindex 5 hat. Transitive und reflexive Kanten sind nicht abgebildet.

Lemma 4.12 *Seien $j \geq i \geq 1$ und $i \neq j - 1$, dann gilt $\hbar(\mathcal{H}_j, i) = i$.*

Die kanonischen Modelle sind die einzigen, bei denen paarweise verschiedene Welten auch immer paarweise verschiedene Modellindizes haben. In diesem Sinne sind sie auch die kleinsten Modelle, denn wenn ein Modell eine Welt mit Modellindex n hat, so muss das Modell nach Konstruktion von $\hbar$ aus mindestens $n - 1$ Welten bestehen.
Der Modellindex einer beliebigen Welt entspricht genau der Welt in einem kanonischen Modell, zu der sie äquivalent ist.

Lemma 4.13 *Es seien $\mathcal{M} \in \mathfrak{K}^{\mathsf{i}}[1]$ und w eine Welt aus $\mathcal{M}$ mit $\hbar(\mathcal{M}, w) = i$. Dann gilt $(\mathcal{M}, w) \equiv_{\mathfrak{F}^{\mathsf{i}}[1]} (\mathcal{H}_j, i)$ für jedes $j \in \{i\} \cup \{i + 2, i + 3, \ldots\}$.*

Beweis. Seien $\mathcal{M} \in \mathfrak{K}^{\mathrm{i}}[1]$ ein Modell, w eine Welt aus $\mathcal{M}$ mit $\hbar(\mathcal{M}, w) = i$ und $j \in \{i\} \cup \{i+2, i+3, \dots\}$. Aus Theorem 4.2 folgt, dass wir nicht für jede Formel $\alpha \in \mathfrak{F}^{\mathrm{i}}[1]$ die Äquivalenz „$\mathcal{M}, w \models \alpha \iff \mathcal{H}_j, i \models \alpha$“ zeigen müssen. Es reicht aus, dies für alle Rieger-Nishimura-Formeln zu prüfen. Des Weiteren ist $(\mathcal{H}_j, i) \equiv_{\mathfrak{F}^{\mathrm{i}}[1]} (\mathcal{H}_i, i)$ klar. Es folgt, dass Lemma 4.13 zur folgenden Behauptung äquivalent ist.

Behauptung 4.2 *Es seien $\mathcal{M}$ ein* IPL[1]*-Modell und w eine Welt aus $\mathcal{M}$. Dann gilt*

$$\mathcal{M}, w \models \alpha \iff \mathcal{H}_{\hbar(\mathcal{M},w)}, \hbar(\mathcal{M}, w) \models \alpha$$

für jede Rieger-Nishimura-Formel α.

Beweis der Behauptung. Wir zeigen diese Behauptung mit Induktion über den Rang von α. Für $\mathcal{M} = (U, \leqslant, \xi, \{p\}) \in \mathfrak{K}^{\mathrm{i}}[1]$, $w \in U$ und eine Rieger-Nishimura-Formel α ist der Fall $\mathit{Rang}(\alpha) \in \{0, 1\}$ klar.
Im Induktionsschritt sei α eine Rieger-Nishimura-Formel mit $\mathit{Rang}(\alpha) = k > 1$. Wir unterscheiden zwei Fälle: Im ersten Fall sei $\alpha = \psi_k$. Da $\psi_k = \varphi_{k-1} \vee \psi_{k-1}$ gilt, folgt die Behauptung direkt aus der Induktionsvoraussetzung. Im zweiten Fall sei $\alpha = \varphi_k$, dann gelten folgende Äquivalenzen:

$$\begin{aligned}
&\mathcal{M}, w \models \varphi_k \quad (= \varphi_{k-1} \rightarrowtriangle \psi_{k-1}) && \text{(i)}\\
\Leftrightarrow\ & \forall v \in U, w \leqslant v : \text{ wenn } \mathcal{M}, v \models \varphi_{k-1}, \text{ dann } \mathcal{M}, v \models \psi_{k-1} && \text{(ii)}\\
\Leftrightarrow\ & \forall v \in U, w \leqslant v : \text{ wenn } \mathcal{H}_{\hbar(\mathcal{M},v)}, \hbar(\mathcal{M}, v) \models \varphi_{k-1}, \\
& \qquad\qquad \text{dann } \mathcal{H}_{\hbar(\mathcal{M},v)}, \hbar(\mathcal{M}, v) \models \psi_{k-1} && \text{(iii)}\\
\Leftrightarrow\ & \forall x \in W_{\hbar(\mathcal{M},w)} : \text{ wenn } \mathcal{H}_x, x \models \varphi_{k-1}, \text{ dann } \mathcal{H}_x, x \models \psi_{k-1} && \text{(iv)}\\
\Leftrightarrow\ & \forall x \in W_{\hbar(\mathcal{M},w)} : \text{ wenn } \mathcal{H}_{\hbar(\mathcal{M},w)}, x \models \varphi_{k-1}, \\
& \qquad\qquad \text{dann } \mathcal{H}_{\hbar(\mathcal{M},w)}, x \models \psi_{k-1} && \text{(v)}\\
\Leftrightarrow\ & \mathcal{H}_{\hbar(\mathcal{M},w)}, \hbar(\mathcal{M}, w) \models \varphi_{k-1} \rightarrowtriangle \psi_{k-1} \quad (= \varphi_k) && \text{(vi)}
\end{aligned}$$

Die Äquivalenz zwischen (i) und (ii) folgt aus der Interpretation von $\rightarrowtriangle$. Aus der Induktionsvoraussetzung folgt die Äquivalenz zwischen (ii) und (iii). Weiter sind (iii) und (iv) äquivalent, weil $\{\hbar(\mathcal{M}, v) \mid v \in U, w \leqslant v\} = \{1, 2, \dots, \hbar(\mathcal{M}, w) - 2\} \cup \{\hbar(\mathcal{M}, w)\} = W_{\hbar(\mathcal{M},w)}$ ist. Da $x \in \{1, 2, \dots, \hbar(\mathcal{M}, w) - 2\} \cup \{\hbar(\mathcal{M}, w)\}$ gilt ($\mathcal{H}_x$ ist ein Teilmodell von $\mathcal{H}_{\hbar(\mathcal{M},w)}$), sind (iv) und (v) äquivalent. Die letzte Äquivalenz zwischen (v) und (vi) beruht wieder auf der Definition von $\rightarrowtriangle$ und der Konstruktion von $\mathcal{H}_{\hbar(\mathcal{M},w)}$ (die Welt $\hbar(\mathcal{M}, w)$ hat keine echten Vorgänger in $\mathcal{H}_{\hbar(\mathcal{M},w)}$). ▪
Mit Theorem 4.2 folgt die in Behauptung 4.2 gezeigte Äquivalenz für alle IPL[1]-Formeln und es gilt $(\mathcal{M}, w) \equiv_{\mathfrak{F}^{\mathrm{i}}[1]} (\mathcal{H}_j, i)$ für $\hbar(\mathcal{M}, w) = i$ und $j \in \{i\} \cup \{i+2, i+3, \dots\}$. □

Die nächste Bemerkung folgt direkt aus Lemma 4.13 bzw. Behauptung 4.2 und kann als Verallgemeinerung von Lemma 4.5 gesehen werden.

Bemerkung 4.14 *Seien* $\alpha \in \mathfrak{F}^{\mathsf{i}}[1] \setminus ([\bot] \cup [\top])$, $\mathcal{M} \in \mathfrak{K}^{\mathsf{i}}[1]$ *und* w *eine Welt aus* $\mathcal{M}$. *Dann gilt*

$$\mathcal{M}, w \models \alpha \iff \begin{cases} \hbar(\mathcal{M}, w) \leq k, & \textit{falls} \quad RNIndex(\alpha) = (k, psi) \\ \hbar(\mathcal{M}, w) < k \text{ } \textit{oder} \\ \hbar(\mathcal{M}, w) = k+1, & \textit{falls} \quad RNIndex(\alpha) = (k, phi) \ . \end{cases}$$

Ob eine (Rieger-Nishimura-)Formel von einer Welt erfüllt ist, hängt also nur von dem Modellindex dieser Welt ab. Deswegen interessiert uns im weiteren Verlauf, wie aufwändig es ist, diesen Modellindex zu bestimmen. Dafür definieren wir folgendes Entscheidungsproblem:

Problem:	ModellIndex
Eingabe:	$\langle \mathcal{M}, w, i \rangle$, wobei $\mathcal{M}$ eine IPL[1]-Modell, w eine Welt aus $\mathcal{M}$ und $i \in \mathbb{N}^+$ ist.
Frage:	Gilt $\hbar(\mathcal{M}, w) = i$?

Will man den Modellindex einer Welt berechnen, kann man dies mit einer alternierenden Turingmaschine tun, indem man die Funktion $\hbar$ direkt implementiert. Dabei ist die Anzahl der Alternierungen durch die Größe des Modells (bzw. die Anzahl der Welten) beschränkt.

Lemma 4.15 *Das Problem* ModellIndex *kann für die Eingabe* $\langle \mathcal{M}, w, i \rangle$ *in* $\mathsf{ALOGSPACE}[\min\{|\mathcal{M}|, i\}]$ *entschieden werden.*

Aus der Definition 4.11 von $\hbar$ kann man sofort ableiten, dass auch i eine obere Schranke für die Zahl der Alternierungen ist. Da $\mathsf{P} = \mathsf{ALOGSPACE}[n^{\mathcal{O}(1)}]$ gilt [CKS81], folgt direkt ModellIndex $\in \mathsf{P}$ und es ergibt sich zusätzlich folgende Bemerkung.

Bemerkung 4.16 *Das Problem* IPL[1]-MÄQ *ist in* P.

Um für zwei Welten aus zwei Modellen IPL[1]-MÄQ zu entscheiden, muss nur geprüft werden, ob der Modellindex beider Welten gleich ist.

4.2.3 Die obere Schranke für die Formelauswertung

Algorithmus 6 entscheidet das Formelauswertungsproblem für IPL[1] auf folgende Weise: Als Eingabe bekommt er eine Formel α, ein Modell $\mathcal{M}$ und eine Welt w aus $\mathcal{M}$. Zuerst wird der Rieger-Nishimura-Index (i, x) von α berechnet. Abhängig von (i, x) ist klar, für welche Modellindizes $\mathcal{M}, w \models \alpha$ gilt (Bemerkung 4.14). Im zweiten Schritt überprüft der Algorithmus dann, ob der Modellindex von w einen dieser Werte annimmt, oder nicht.

Theorem 4.17 *Das Problem* IPL[1]-FA *ist in* AC^1.

Beweis. Zuerst zeigen wir, dass Algorithmus 6 das Formelauswertungsproblem entscheidet. Anschließend analysieren wir seine Komplexität.
Der Algorithmus bekommt die Eingabe $\langle \alpha, \mathcal{M}, w \rangle$ und akzeptiert diese, grob gesagt, genau dann, wenn der Rieger-Nishimura-Index $\mathit{RNIndex}(\alpha)$ von α und der Modellindex $\hbar(\mathcal{M}, w)$ von w in $\mathcal{M}$ gemäß Bemerkung 4.14 zusammenpassen. Der Rieger-Nishimura-Index von α wird in den Zeilen 1 und 2 berechnet (bzw. nichtdeterministisch geraten). Die trivialen Fälle (α hat den Rang 0 und ist äquivalent zu einer Konstanten) werden in den Zeilen 3 und 4 behandelt. Nach Bemerkung 4.14 gilt für eine Rieger-Nishimura-Formel β_i mit $\mathit{Rang}(\beta_i) = i > 0$ das Folgende: Entweder ist $\beta_i = \psi_i$ (Zeile 5). Dann gilt $\mathcal{M}, w \models \beta_i$ genau dann, wenn $\hbar(\mathcal{M}, w) \leq i$ ist. Dieser Fall wird in Zeile 6 überprüft. Oder $\beta_i = \varphi_i$ (Zeile 8). Dann gilt $\mathcal{M}, w \models \beta_i$ genau dann, wenn $\hbar(\mathcal{M}, w) = i + 1$ oder $\hbar(\mathcal{M}, w) < i$ ist. Dieser Fall wird in Zeile 9 überprüft. Wenn $\hbar(\mathcal{M}, w) > \mathit{Rang}(\alpha) + 1$ ist, dann gilt $\mathcal{M}, w \not\models \alpha$ (Bemerkung 4.14). Damit ist gezeigt, dass Algorithmus 6 korrekt arbeitet.

Algorithmus 6 `Formelauswertung für` IPL[1]

Eingabe: Formel $\alpha \in \mathfrak{F}^{\mathsf{i}}[1]$, Modell $\mathcal{M} \in \mathfrak{K}^{\mathsf{i}}[1]$, Welt w aus $\mathcal{M}$

```
 1: rate nichtdet. einen Rieger-Nishimura-Index (i, x) mit i ≤ c · log(|α|)
 2: wenn ⟨α, (i, x)⟩ ∈ ÄqRN-Formel dann
 3:   wenn (i, x) = (0, ⊥) dann lehne ab
 4:   wenn (i, x) = (0, ⊤) dann akzeptiere
 5:   wenn x = psi dann
 6:     wenn ħ(M, w) ∈ {1, 2, ..., i} dann akzeptiere
 7:       sonst lehne ab
 8:   wenn x = phi dann
 9:     wenn ħ(M, w) ∈ {1, 2, ..., i − 1} ∪ {i + 1} dann akzeptiere
10:       sonst lehne ab
11:   sonst lehne ab
```

Im Weiteren untersuchen wir die Komplexität von Algorithmus 6. In Zeile 1 wird nichtdeterministisch ein Rieger-Nishimura-Index (i, x) geraten. Die Entscheidung in Zeile 2, ob $\langle \alpha, (i, x) \rangle \in$ ÄqRN-Formel gilt, kann gemäß Lemma 4.8 mit den Ressourcen von $\mathsf{LOGdetCFL}$ gefällt werden. Für ein Modell $\mathcal{M}$, eine Welt w aus $\mathcal{M}$ und eine Zahl n kann die Frage, ob $\hbar(\mathcal{M}, w) = n$ gilt, in $\mathsf{ALOGSPACE}[n]$ beantwortet werden (Lemma 4.15). Die Alternierungstiefe bei dieser Entscheidung ist durch den zu überprüfenden Modellindex beschränkt. Daher kann die Entscheidung aus Zeile 6 (bzw. Zeile 9), ob $\hbar(\mathcal{M}, s) \in \{1, 2, \ldots, i\}$ (bzw. $\hbar(\mathcal{M}, s) \in \{1, 2, \ldots, i-1\} \cup \{i+1\}$) gilt, mit i (bzw. $i+1$) Alternierungen gefällt werden. Aus Lemma 4.7 ist bekannt, dass i (der Rang von α) höchstens $c \cdot \log(|\alpha|)$ sein kann, wobei c konstant und unabhängig von α ist. Also können die Entscheidungen aus den Zeilen 6 und 9 mit den Ressourcen von $\mathsf{ALOGSPACE}[\log(|\langle \alpha, \mathcal{M}, w \rangle|)]$ getroffen werden. Während der

ganzen Berechnung speichert der Algorithmus nur eine konstante Zahl von Rieger-Nishimura-Indizes und Modellindizes. Da die Größe dieser Indizes logarithmisch beschränkt ist (Lemma 4.7 und $\hbar(\mathcal{M}, w) \leq |\mathcal{M}|$), benötigt der Algorithmus nur logarithmischen Speicherplatz. Aus $\mathsf{LOGdetCFL} \subseteq \mathsf{AC}^1 = \mathsf{ALOGSPACE}[\log]$ [Coo85] folgt die gewünschte obere Schranke: IPL[1]-FA $\in \mathsf{AC}^1$. □

4.3 Untere Schranken

Das Hauptresultat dieses Abschnittes ist die AC^1-Härte von IPL[1]-FA. Dafür geben wir eine Reduktion von $\textsc{Asgep}_{\log}$ auf IPL[1]-FA an. Diese Reduktion basiert wesentlich auf einer Transformation, die alternierende Schichtgraphen (logarithmischer Tiefe) in IPL[1]-Modelle überführt (siehe Abbildung 4.4).
Zur Erinnerung, ein alternierender Schichtgraph besteht aus Schichten von Knoten. Kanten gibt es nur zwischen direkt benachbarten Schichten. Die Schichten mit gerader Nummer sind die ∀-Schichten und die mit ungerader Nummer die ∃-Schichten. Ein alternierender Pfad beginnt in einem Knoten in der untersten Schicht und endet in einem der obersten Schicht. Ist ein Knoten einer ∃-Schicht Bestandteil dieses Pfades, so muss wenigstens einer der Nachbarknoten (aus der nächsten Schicht) ebenfalls ein Bestandteil sein. Für Knoten aus den ∀-Schichten gilt, sind sie Bestandteil des Pfades, so müssen alle Nachbarknoten auch Teil des Pfades sein. Ein Beispiel ist in Abbildung 2.2 (bzw. Abbildung 4.4) zu sehen, formal sind alternierende Pfade in Definition 2.18 und alternierende Schichtgraphen in Definition 2.20 definiert. Asgep – die Frage, ob es einen alternierenden Pfad von einem Start- zu einem Zielknoten gibt – ist P-vollständig (Theorem 2.21). Für die Variante $\textsc{Asgep}_{\log}$ – der Graph hat nur logarithmisch viele Schichten – wird in Theorem 2.22 die AC^1-Vollständigkeit gezeigt.

4.3.1 Alternierende Schichtgraphen und Kripke-Modelle

Wir geben jetzt eine Konstruktion an, mit der man eine Asgep-Instanz $\langle G, s, t\rangle$ in ein Modell $\mathcal{M}_G$ umwandelt. Die alternierenden Schichtgraphen wurden in Definition 2.20 so eingeführt, dass die Schichten von „unten“ nach „oben“ beginnend mit Eins nummeriert wurden. Kanten verlaufen demnach immer von Schicht i zu Schicht $i + 1$. In der folgenden Konstruktion ist es aus technischen Gründen einfacher, die Schichten in umgekehrter Reihenfolge zu nummerieren.
Sei $\langle G, s, t\rangle$ eine Instanz von Asgep mit $G = (V_G, E_G)$, $V_G = V_\exists \cup V_\forall$ und den m Schichten $V_\exists = V_{m-1} \cup V_{m-3} \cup \cdots \cup V_1$ und $V_\forall = V_{m-2} \cup V_{m-4} \cup \cdots \cup V_0$. Für alle Schichten V_i mit $m - 1 \geq i > 0$ gilt, dass Kanten nur zwischen V_i und V_{i-1} verlaufen. Wir konstruieren jetzt ein Modell $\mathcal{M}_G = (W, \leqslant, \xi, \{p\})$ aus dieser Instanz. Ein Beispiel dieser Konstruktion ist in Abbildung 4.4 zu sehen.
Wir geben für jedes $i = 0, 1, 2, \ldots, m - 1$ zwei Mengen von Welten an:

$$\begin{aligned} U_i^{in} &:= \{v^{in} \mid v \in V_i\} \ , \\ U_i^{out} &:= \{v^{out} \mid v \in V_i\} \ . \end{aligned}$$

Die Menge aller in diesen Mengen vorkommenden Welten bezeichnen wir mit U:

$$U := \bigcup_{i=0}^{m-1} (U_i^{in} \cup U_i^{out}) \ .$$

Die Knoten aus den *in*-Schichten sind *Eingangsknoten* und die aus den *out*-Schichten sind *Ausgangsknoten*. Jede Kante (u, v) aus E_G wird zu einer Kante (u^{out}, v^{in}) transformiert. Jeder Knoten einer *in*-Schicht hat einen korrespondierenden Knoten in der zugehörigen *out*-Schicht, mit dem er verbunden ist. Damit ergibt sich die folgende Kantenmenge E:

$$E := \big\{(u^{out}, v^{in}) \mid (u, v) \in E_G\big\} \cup \big\{(v^{in}, v^{out}) \mid v \in V_G\big\} \ .$$

Sei $G' = (U, E)$ der Graph, der aus G durch die obige Konstruktion entsteht. Wenn wir für $u \in V_\exists$ (bzw. $u \in V_\forall$) die Knoten $u^x \in U$ für $x \in \{in, out\}$ als $\exists$-Knoten (bzw. $\forall$-Knoten) auffassen, dann gilt $aPfad_G(v, w)$ für Knoten $v, w \in V_G$ genau dann, wenn auch $aPfad_{G'}(v^{in}, w^{out})$ gilt. Es ist klar, dass G' kein alternierender Schichtgraph im ursprünglichen Sinne ist. Bei dieser Konstruktion folgen immer zwei $\exists$- bzw. zwei $\forall$-Schichten aufeinander, da jede Schicht aus dem originalen Graph G in eine *in*- und eine *out*-Schicht transformiert wurde.
Im nächsten Schritt fügen wir das kanonische Modell $\mathcal{H}_{4m-2} = (\{1, 2, \ldots, 4m - 4\} \cup \{4m - 2\}, \trianglelefteq, \xi_{4m-2}, \{p\})$ zu G' hinzu. Dabei werden die Welten 1 und 2 in Schicht U_0^{out} eingefügt, die Welten 3 und 4 in Schicht U_0^{in}, die Welten 5 und 6 in Schicht U_1^{in} und so weiter. Für $i = 0, 1, 2, \ldots, m - 2$ seien

$$\begin{aligned} W_i^{out} &:= U_i^{out} \cup \{4i + 1, 4i + 2\} \ , \\ W_i^{in} &:= U_i^{in} \cup \{4i + 3, 4i + 4\} \ , \\ W_{m-1}^{out} &:= U_{m-1}^{out} \cup \{4m - 2\} \ \text{ und} \\ W_{m-1}^{in} &:= U_{m-1}^{in} \ . \end{aligned}$$

Wir können jetzt die Menge W aller Welten aus $\mathcal{M}_G$ angeben:

$$W := \bigcup_{i=0}^{m-1} (W_i^{out} \cup W_i^{in}) \ .$$

Jetzt fügen wir die Kanten des kanonischen (Teil-)Modells $\mathcal{H}_{4m-2}$ ein:

$$\begin{aligned} H := \ & \{(i, i-2) \mid i \in \{3, 4, \ldots, 4m - 4\} \cup \{4m - 2\}\} \ \cup \\ & \{(i, i-3) \mid i \in \{4, 5, \ldots, 4m - 2\}\} \ . \end{aligned}$$

Die Menge H enthält genau die Kanten aus $\mathcal{H}_{4m-2}$, die diesem kanonischen Modell seine typische Struktur geben. Jeder Knoten i hat $i-2$ und $i-3$, aber nicht $i-1$ als Vorgänger und die Sichtbarkeitsrelation $\trianglelefteq$ ist die transitive und reflexive Hülle

von H. Als nächstes verbinden wir die Knoten aus U mit den Knoten aus $\mathcal{H}_{4m-2}$:

$$\begin{aligned} T_{in} &:= \{(u, 4i+2) \mid u \in U_i^{in}, i = 0, 1, 2, \ldots, m-1\} , \\ T_{out} &:= \{(u, 4i-1) \mid u \in U_i^{out}, i = 1, 2, \ldots, m-1\} . \end{aligned}$$

Wir haben jetzt den Graphen $(W, E \cup H \cup T_{in} \cup T_{out})$ konstruiert. Für die Nutzung dieses Graphen als Rahmen für ein IPL[1]-Modell fehlen noch Transitivität und Reflexivität. Die Reduktionsfunktion, die einen alternierenden Schichtgraph in ein IPL[1]-Modell überführt, muss in logarithmischem Platz berechenbar sein. Wegen dieser Beschränkung können wir nicht einfach die transitive Hülle verwenden, sondern arbeiten mit der pseudotransitiven Hülle (siehe Definition 2.23):

$$\begin{aligned} S^{trans} := \bigcup_{i=m-1}^{1} \Bigg[\Big(W_i^{in} \times \bigcup_{j=i-1}^{0} W_j^{in} \cup W_j^{out} \Big) \cup \\ \Big(W_i^{out} \times \big(W_{i-1}^{out} \cup \bigcup_{j=i-2}^{0} W_j^{in} \cup W_j^{out} \big) \Big) \Bigg] . \end{aligned}$$

Es fehlen noch die reflexiven Kanten:

$$S^{refl} := \{(w, w) \mid w \in W\} .$$

Die Sichtbarkeitsrelation $\leqslant$ für $\mathcal{M}_G$ ist jetzt wie folgt gegeben:

$$\leqslant := E \cup H \cup T_{in} \cup T_{out} \cup S^{trans} \cup S^{refl} .$$

Abschließend geben wir die Belegungsfunktion ξ von $\mathcal{M}_G$ an:

$$\xi(p) := \{t^{out}, 1\} .$$

Dabei ist $t^{out} \in U_0^{out}$ die Kopie des Zielknotens t und 1 der Knoten aus $\mathcal{H}_{4m-2}$ mit $\xi_{4m-2}(p) = \{1\}$. Damit ist das IPL[1]-Modell $\mathcal{M}_G = (W, \leqslant, \xi, \{p\})$ konstruiert Ein Beispiel einer ASGEP-Instanz $\langle G, s, t\rangle$ und dem korrespondierenden IPL[1]-Modell $\mathcal{M}_G$ ist in Abbildung 4.4 zu sehen.
Die Verdopplung der Schichten in *in*- und *out*-Schichten und die Hinzunahme des kanonischen Modells benötigen wir, um den Welten in verschiedenen Schichten einen eindeutigen Modellindex zu geben – je nachdem, ob sie Bestandteil des alternierenden Pfades von s nach t sind oder nicht. Der Zusammenhang zwischen den Modellindizes in $\mathcal{M}_G$ und der *aPfad*-Eigenschaft wird in folgendem Lemma angegeben.

Lemma 4.18 *Es sei $i = 0, 1, 2, \ldots, m-1$ und $v \in V_i$. Dann gilt*

(1) *wenn i gerade ist ($\forall$-Schicht):*

$$\hbar(\mathcal{M}_G, v^{in}) = \begin{cases} 4i+4, & \textit{falls} \quad aPfad_G(v,t) & \text{(i)} \\ 4i+2, & \textit{sonst} & \text{(ii)} \end{cases}$$

$$\hbar(\mathcal{M}_G, v^{out}) = \begin{cases} 4i+1, & \textit{falls} \quad aPfad_G(v,t) & \text{(iii)} \\ 4i+2, & \textit{sonst} & \text{(iv)} \end{cases}$$

und

(2) *wenn i ungerade ist ($\exists$-Schicht):*

$$\hbar(\mathcal{M}_G, v^{in}) = \begin{cases} 4i+2, & \textit{falls} \quad aPfad_G(v,t) & \text{(v)} \\ 4i+4, & \textit{sonst} & \text{(vi)} \end{cases}$$

$$\hbar(\mathcal{M}_G, v^{out}) = \begin{cases} 4i+2, & \textit{falls} \quad aPfad_G(v,t) & \text{(vii)} \\ 4i+1, & \textit{sonst}\ . & \text{(viii)} \end{cases}$$

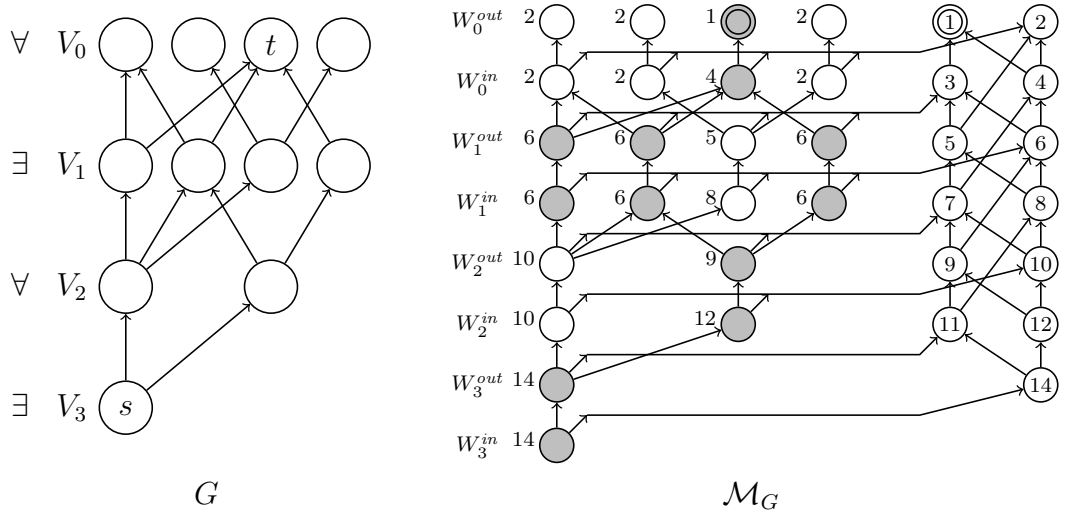

Abbildung 4.4: Ein alternierender Schichtgraph G (links) und das transformierte Modell $\mathcal{M}_G$ (rechts). Die Welten, in denen p erfüllt ist, sind doppelt umrandet, transitive und reflexive Kanten in $\mathcal{M}_G$ sind nicht abgebildet. Der Wert x links einer Welt aus $\mathcal{M}_G$ ist ihr Modellindex $\hbar(\mathcal{M}_G, x)$. Die Namen der Welten aus $\mathcal{H}_{14}$ und deren Modellindizes sind identisch und stehen in den Welten. Welten w^{in} und w^{out} sind grau gefärbt, wenn $aPfad_G(w,t)$ in G gilt.

Beweis. Den Beweis führen wir mit vollständiger Induktion über die Schichten von $\mathcal{M}_G$. Wichtig ist, dass $\hbar$ die Identität auf den Welten des eingebetteten Modells $\mathcal{H}_{4m-2}$ ist, da diese Welten keine ausgehenden Kanten zu Welten aus U (außerhalb von $\mathcal{H}_{4m-2}$) haben. Es gilt also $\hbar(\mathcal{M}_G, j) = j$ für alle $j \in W \setminus U$.
Für den Induktionsanfang sei $v \in U_0^{out}$. Dann gilt $\hbar(\mathcal{M}_G, v) = 1$, wenn $v = t^{out}$, und $\hbar(\mathcal{M}_G, v) = 2$, wenn $v \neq t^{out}$ ist.
Im Induktionsschritt schauen wir uns die weiteren Schichten an. Wir führen den Beweis hier für gerade i ($\forall$-Schicht), dabei zeigen wir die Fälle (i) bis (iv) separat. Der Beweis für ungerade i verläuft dann nach demselben Prinzip.

(i) Sei $v^{in} \in U_i^{in}$ für gerade i (∀-Schicht) und es gelte $aPfad_G(v,t)$. Wegen $v^{in} \leqslant 4i+2$ und weil $\hbar(\mathcal{M}_G, 4i+2) = 4i+2$ ist, gilt

$$\exists u_1 \in W_{v^{in}\uparrow} : \hbar(\mathcal{M}_G, u_1) = 4i+2 \ .$$

Aus der Induktionsvoraussetzung folgt $\hbar(\mathcal{M}_G, v^{out}) = 4i+1$. Nach Konstruktion gilt $v^{in} \leqslant v^{out}$ und es folgt

$$\exists u_2 \in W_{v^{in}\uparrow} : \hbar(\mathcal{M}_G, u_2) = 4i+1 \ .$$

Zusätzlich folgt aus der Induktionsvoraussetzung und der Konstruktion von $\mathcal{M}_G$, dass keine Welt aus $W_{v^{in}\uparrow}$ den Modellindex $4i+3$ hat, also gilt auch

$$\forall w \in W_{v^{in}\uparrow} : \hbar(\mathcal{M}_G, w) \neq 4i+3 \ .$$

Nach Definition 4.11 folgt damit $\hbar(\mathcal{M}_G, v^{in}) = 4i+4$.

(ii) Sei $v^{in} \in U_i^{in}$ für gerade i (∀-Schicht) und $aPfad_G(v,t)$ gelte nicht. Weil $v^{in} \leqslant 4i+2$ und $\hbar(\mathcal{M}_G, 4i+2) = 4i+2$ ist, gilt nach Definition 4.11

$$\exists u_1, u_2 \in W_{v^{in}\uparrow} : \hbar(\mathcal{M}_G, u_1) = 4i \text{ und } \hbar(\mathcal{M}_G, u_2) = 4i-1 \ .$$

Aus der Induktionsvoraussetzung und der Konstruktion von $\mathcal{M}_G$ folgt für alle Welten in $W_{v^{in}\uparrow}$, dass ihr Modellindex nicht $4i+1$ ist, also

$$\forall w \in W_{v^{in}\uparrow} : \hbar(\mathcal{M}_G, w) \neq 4i+1 \ .$$

Damit folgt $\hbar(\mathcal{M}_G, v^{in}) = 4i+2$ nach Definition 4.11.

(iii) Sei $v^{out} \in U_i^{out}$ für gerade i (∀-Schicht) und es gelte $aPfad_G(v,t)$. Wegen $aPfad_G(v,t)$ folgt aus der Induktionsvoraussetzung für alle $w^{in} \in U_{i-1}^{in}$ mit $v^{out} \leqslant w^{in}$, dass $\hbar(\mathcal{M}_G, w^{in}) = 4i-2$ gilt. Damit erhalten wir

$$\exists u_1 \in W_{v^{out}\uparrow} : \hbar(\mathcal{M}_G, u_1) = 4i-2 \ .$$

Nach Konstruktion von $\mathcal{M}_G$ ist $v^{out} \leqslant 4i-1$ und $\hbar(\mathcal{M}_G, 4i-1) = 4i-1$ und es folgt

$$\exists u_2 \in W_{v^{out}\uparrow} : \hbar(\mathcal{M}_G, u_2) = 4i-1 \ .$$

Des Weiteren folgt aus der Induktionsvoraussetzung und der Konstruktion von $\mathcal{M}_G$, dass es in $W_{v^{out}\uparrow}$ keine Welt mit Modellindex $4i$ gibt, also

$$\forall w \in W_{v^{out}\uparrow} : \hbar(\mathcal{M}_G, w) \neq 4i \ .$$

Nach Definition 4.11 gilt damit $\hbar(\mathcal{M}_G, v^{out}) = 4i+1$.

(iv) Sei $v^{out} \in U_i^{out}$ für gerade i (∀-Schicht) und $aPfad_G(v,t)$ gelte nicht. Nach Konstruktion von $\mathcal{M}_G$ ist $v^{out} \leqslant 4i-1$ und $\hbar(\mathcal{M}_G, 4i-1) = 4i-1$, also gilt

$$\exists u_1 \in W_{v^{out}\uparrow} : \hbar(\mathcal{M}_G, u_1) = 4i-1 \ .$$

Weil $aPfad_G(v,t)$ nicht gilt, gibt es eine Welt $w^{in} \in U_{i-1}^{in}$ mit $v^{out} \leqslant w^{in}$, für die nach Induktionsvoraussetzung $\hbar(\mathcal{M}_G, w^{in}) = 4i$ folgt. Damit gilt

$\exists u_2 \in W_{v^{out}\uparrow} : \hbar(\mathcal{M}_G, u_2) = 4i$.

Zusätzlich folgt aus der Induktionsvoraussetzung und der Konstruktion von $\mathcal{M}_G$, dass keine Welt aus $W_{v^{out}\uparrow}$ den Modellindex $4i + 1$ hat, also

$\forall w \in W_{v^{out}\uparrow} : \hbar(\mathcal{M}_G, w) \neq 4i + 1$.

Und nach Definition 4.11 gilt $\hbar(\mathcal{M}_G, v^{out}) = 4i + 2$.

Der Beweis für die Fälle (v) bis (viii) für ungerade i ($\exists$-Schichten) verläuft analog zu den hier gezeigten Fällen. □

Wir bezeichnen mit g die Funktion, die eine ASGEP-Instanz $\langle G, s, t\rangle$ auf das Modell $\mathcal{M}_G = g(\langle G, s, t\rangle)$ abbildet. Die folgenden Aussagen über g lassen sich unmittelbar aus der Konstruktion von $\mathcal{M}_G$ ableiten.

Bemerkung 4.19 *Für g gelten folgende Eigenschaften:*

(1) *Die Funktion g ist in logarithmischem Platz berechenbar.*

(2) *Für $\langle G, s, t\rangle$, wobei G aus n Knoten und $m \leq n$ Schichten besteht, ist $g(\langle G, s, t\rangle)$ ein* IPL[1]*-Modell mit $\leq 2n + 4m - 3$ Welten und der Tiefe $2m$.*

Die Funktion g ist ein wesentlicher Bestandteil unserer Reduktion für das Härteresultat von IPL[1]-FA. Vorher können wir noch mit Hilfe von g eine untere Schranke für die Berechnung von Modellindizes angeben.

Lemma 4.20 *Das Problem* MODELLINDEX *ist* P*-hart.*

Beweis. Wir geben eine Reduktion von dem P-harten Problem ASGEP an. Für die ASGEP-Instanz $\langle G, s, t\rangle$ mit dem aus m Schichten bestehenden alternierenden Schichtgraph G sei $\mathcal{M}_G = g(\langle G, s, t\rangle)$. Nach Bemerkung 4.19(1) ist g in logarithmischem Platz berechenbar. Aus der Konstruktion von $\mathcal{M}_G$ folgt $\hbar(\mathcal{M}_G, s^{out}) \in \{4(m-1)+1, 4(m-1)+2\}$ und aus Lemma 4.18 folgt, dass $aPfad_G(s, t)$ genau dann gilt, wenn $\hbar(\mathcal{M}_G, s^{out}) = 4m - 2$ ist, also

$$\langle G, s, t\rangle \in \text{ASGEP} \iff \hbar(\mathcal{M}_G, s^{out}) = 4m - 2 \ .$$

Mit Theorem 2.21 folgt die P-Härte von MODELLINDEX. □

Aus $\hbar(\mathcal{M}_G, s^{out}) \in \{4m - 3, 4m - 2\}$ folgt sogar, dass bereits die Berechnung des letzten Bits eines Modellindexes P-hart ist. Analog zur oberen Schranke von IPL[1]-MÄQ aus Bemerkung 4.16 können wir aus Lemma 4.20 auch eine untere Schranke für dieses Problem ableiten.

Bemerkung 4.21 *Das Problem* IPL[1]-MÄQ *ist* P*-hart.*

4.3.2 Die untere Schranke für die Formelauswertung

Im Beweis von Lemma 4.20 zeigten wir, dass es in einem Schichtgraphen G mit m Schichten genau dann einen alternierenden Pfad von s nach t gibt, wenn in dem korrespondierenden Modell $\mathcal{M}_G$ die Welt s^{out} den Modellindex $4m-2$ hat. Aus der Konstruktion von $\mathcal{M}_G$ folgt, dass $\hbar(\mathcal{M}_G, s^{out}) \in \{4m-2, 4m-3\}$ ist und nach Bemerkung 4.14 gilt

$$\hbar(\mathcal{M}_G, s^{out}) = 4m-2 \iff \mathcal{M}_G, s^{out} \models \varphi_{4m-3} \ .$$

In Lemma 4.7 zeigten wir, dass die Länge von φ_{4m-3} exponentiell in m ist. Eine Abbildung von $\langle G, s, t\rangle$ auf $\langle \varphi_{4m-3}, g(\langle G, s, t\rangle), s^{out}\rangle$ kann also nicht in logarithmischem Platz berechnet werden. Wählt man hingegen G so, dass die Zahl m der Schichten logarithmisch in der Größe von G ist, dann ist die Länge der Rieger-Nishimura-Formel φ_{4m-3} polynomiell in der Größe von G und die Abbildung kann in logarithmischem Platz berechnet werden. Die $\text{ASGEP}_{\log}$-Instanzen erfüllen genau diese Einschränkung und wir können $\text{ASGEP}_{\log}$ auf IPL[1]-FA reduzieren.

Theorem 4.22 *Das Problem* IPL[1]-FA *ist* AC^1*-hart.*

Beweis. Zum Beweis reduzieren wir $\text{ASGEP}_{\log}$ auf IPL[1]-FA. Aus Theorem 2.22 folgt, dass $\text{ASGEP}_{\log}$ AC^1-vollständig ist. Seien $\langle G_{\log}, s, t\rangle$ eine $\text{ASGEP}_{\log}$-Instanz mit m Schichten und $\mathcal{M}_{G_{\log}} = g(\langle G_{\log}, s, t\rangle)$. Wir definieren folgende Reduktionsfunktion r für $\text{ASGEP}_{\log}$-Instanzen auf IPL[1]-FA-Instanzen:

$$r(\langle G_{\log}, s, t\rangle) := \langle \varphi_{4m-3}, \mathcal{M}_{G_{\log}}, s^{out}\rangle \ .$$

Da $\langle G_{\log}, s, t\rangle$ auch eine ASGEP-Instanz ist, kann die Funktion g, die dieser Instanz das IPL[1]-Modell $\mathcal{M}_{G_{\log}}$ zuordnet, in logarithmischem Platz berechnet werden (Bemerkung 4.19(1)). Nach Definition ist m logarithmisch in der Größe von $G_{\log}$ und die Länge der Rieger-Nishimura-Formel φ_{4m-3} ist somit polynomiell in der Größe von $G_{\log}$. Das Modell $\mathcal{M}_{G_{\log}}$ ist etwa genauso groß wie $G_{\log}$ (siehe dazu Bemerkung 4.19(2)). Damit folgt, dass die Reduktion r in logarithmischem Platz berechnet werden kann. Wir haben im Beweis von Lemma 4.20 bereits

$$aPfad_{G_{\log}}(s,t) \iff \hbar(\mathcal{M}_{G_{\log}}, s^{out}) = 4m-2$$

gezeigt. Wegen Bemerkung 4.14 und der Konstruktion von $\mathcal{M}_{G_{\log}}$ gilt

$$\hbar(\mathcal{M}_{G_{\log}}, s^{out}) = 4m-2 \iff \mathcal{M}_{G_{\log}}, s^{out} \models \varphi_{4m-3} \ ,$$

damit folgt sofort die Korrektheit von r

$$\langle G_{\log}, s, t\rangle \in \text{ASGEP}_{\log} \iff \langle \varphi_{4m-3}, \mathcal{M}_{G_{\log}}, s^{out}\rangle \in \text{IPL[1]-FA}$$

und IPL[1]-FA ist AC^1-hart. □

Da $\wedge$ in φ_{4m-3} nicht vorkommt, ist bereits IPL$[\bot, \vee, \rightarrow, 1]$-FA AC^1-hart.

4.4 Superintuitionistische Logiken mit einer Variablen

In superintuitionistischen Logiken gibt es mehr Tautologien als in IPL. Wir betrachten jetzt die Fragmente, in denen nur eine Variable vorkommt. Im syntaktischen Sinne (in Bezug auf das natürliche Schließen) lässt sich jede dieser Logiken aus IPL[1] durch Hinzunahme einer Formel aus $\mathfrak{F}^{\mathsf{i}}[1] \setminus ([\bot] \cup [\top])$ als Axiom erzeugen. Das bedeutet, dass es für jede Rieger-Nishimura-Formel (außer $\bot$ und $\top$) genau eine solche Logik gibt (und auch sonst keine weiteren). Wir bezeichnen diese Logiken mit IPL[1]$^{\alpha}$, wobei $\alpha \in \{\varphi_1, \psi_1, \varphi_2, \psi_2, \dots\}$ das zusätzliche Axiom ist. Semantisch bedeutet die Hinzunahme einer Formel α als Axiom, dass alle Formeln der Logik IPL[1]$^{\alpha}$ nur noch über den Modellen interpretiert werden, in denen α in jeder Welt erfüllt ist. Wir werden zeigen, dass diese Einschränkung der zugelassenen Modelle dazu führt, dass die Logik IPL[1]$^{\alpha}$ endlich erzeugt ist und somit die Formelauswertung in NC^1 möglich ist.

Theorem 4.23 *Für alle Rieger-Nishimura-Formeln* $\alpha \in \{\varphi_1, \psi_1, \varphi_2, \psi_2, \dots\}$ *ist das Problem* IPL[1]$^{\alpha}$-FA *in* NC^1.

Beweis. Wir zeigen, dass IPL[1]$^{\alpha}$ endlich erzeugt ist, mit Theorem 3.3 folgt dann sofort, dass das Formelauswertungsproblem in NC^1 ist.

Behauptung 4.3 *Für* $\alpha \in \{\varphi_1, \psi_1, \varphi_2, \psi_2, \dots\}$ *ist* IPL[1]$^{\alpha}$ *endlich erzeugt.*

Beweis der Behauptung. Für $i > 0$ unterscheiden wir die beiden Fälle $\alpha = \psi_i$ und $\alpha = \varphi_i$. Für $\alpha = \psi_i$ ist die Logik IPL[1]$^{\psi_i} = (\mathfrak{F}^{\mathsf{i}}[1], \mathfrak{K}^{\mathsf{i}}[1]^{\psi_i})$. Seien $\mathcal{M} \in \mathfrak{K}^{\mathsf{i}}[1]$ ein Modell und w eine Welt aus $\mathcal{M}$. Nach Bemerkung 4.14 gilt genau dann $\mathcal{M}, w \models \psi_i$, wenn $\hbar(\mathcal{M}, w) \leq i$ ist. Damit folgt sofort $\mathfrak{K}^{\mathsf{i}}[1]^{\psi_i} = \{\mathcal{M} = (W, \leqslant, \xi, \{p\}) \mid \forall w \in W : \hbar(\mathcal{M}, w) \leq i\}$. Da der Modellindex einer Welt in einem Modell aus $\mathfrak{K}^{\mathsf{i}}[1]^{\psi_i}$ den Wert i nicht übersteigt, gibt es bezüglich $\equiv_{\mathfrak{F}^{\mathsf{i}}[1]}$ nur endlich viele Äquivalenzklassen. Damit folgt aus Lemma 3.6, dass IPL[1]$^{\psi_i}$ endlich erzeugt ist. Analog kann man zeigen, dass jede Welt eines IPL[1]$^{\varphi_i}$-Modells einen Modellindex aus $\{1, 2, \dots, i-1\} \cup \{i+1\}$ hat. Damit folgt aus Lemma 3.6, dass auch IPL[1]$^{\varphi_i}$ endlich erzeugt ist. ▪

Mit Behauptung 4.3 und Theorem 3.3 folgt IPL[1]$^{\alpha}$-FA $\in \mathsf{NC}^1$. □

Das Formelauswertungsproblem für Formeln ohne Variablen über Modellen mit nur einer, sich selbst sehenden Welt ist bereits NC^1-hart (Theorem 2.31). Diese Formeln und dieses Modell sind Bestandteil jeder superintuitionistischen Logik. Damit können wir folgende untere Schranke angeben.

Theorem 4.24 *Für alle Rieger-Nishimura-Formeln* $\alpha \in \{\varphi_1, \psi_1, \varphi_2, \psi_2, \dots\}$ *ist das Problem* IPL[1]$^{\alpha}$-FA NC^1*-hart.*

Aus dem Beweis von Behauptung 4.3 wissen wir bereits, welche Modellklassen es in der Logik $\mathrm{IPL}[1]^{\alpha} = (\mathfrak{F}^{\mathsf{i}}[1], \mathfrak{K}^{\mathsf{i}}[1]^{\alpha})$ für $\alpha \in \{\varphi_1, \psi_1, \varphi_2, \psi_2, \dots\}$ gibt. Konkret gilt für den Modellindex einer Welt w eines Modells $\mathcal{M} \in \mathfrak{K}^{\mathsf{i}}[1]^{\alpha}$

$$\hbar(\mathcal{M}, w) \quad \in \quad \begin{cases} \{1, 2, \dots, n-1\} \cup \{n+1\}, & \text{falls} \quad \alpha = \varphi_n \\ \{1, 2, \dots, n\}, & \text{falls} \quad \alpha = \psi_n \ . \end{cases}$$

Jetzt wollen wir noch klären, welche Formeläquivalenzklassen es gibt. Welche Rieger-Nishimura-Formel in welchem Modell (bzw. in dessen Basiswelt) erfüllt ist, geht aus Lemma 4.5 hervor und wird in Tabelle 4.2 dargestellt. Für die Zerlegung von $\mathfrak{F}^{\mathsf{i}}[1]$ in Formelklassen in $\mathrm{IPL}[1]^{\alpha}$ ergibt sich

$$\mathfrak{F}^{\mathsf{i}}[1] \quad = \quad \begin{cases} [\bot] \cup [\varphi_1] \cup [\psi_1] \cup \dots \cup [\varphi_{n-1}] \cup [\psi_{n-1}] \cup [\varphi_n], & \text{falls} \quad \alpha = \varphi_n \\ [\bot] \cup [\varphi_1] \cup [\psi_1] \cup \dots \cup [\varphi_{n-1}] \cup [\psi_{n-1}] \cup [\psi_n], & \text{falls} \quad \alpha = \psi_n \ . \end{cases}$$

Interessanterweise gilt in den Logiken $\mathrm{IPL}[1]^{\psi_n}$, dass ψ_{n-1} und φ_n äquivalent sind, da $\mathcal{H}_{n+1}$ das einzige Modell ist, in dem sie sich unterscheiden, aber dieses Modell in $\mathfrak{K}^{\mathsf{i}}[1]^{\psi_n}$ nicht vorkommt. Demnach enthalten die Klassen $[\psi_{n-1}]$ und $[\top]$ in der Logik $\mathrm{IPL}[1]^{\psi_n}$ andere Formeln, als in der Logik $\mathrm{IPL}[1]^{\varphi_n}$. Alle anderen Klassen sind in beiden Logiken identisch. Aus ähnlichen Gründen sind in $\mathrm{IPL}[1]^{\varphi_n}$ für $n \geq 1$ die Formeln ψ_{n-1} und ψ_n äquivalent. Die Struktur der Verbände für ungerade n zeigt Abbildung 4.5. Für gerade n sind die Verbände sehr ähnlich aufgebaut.

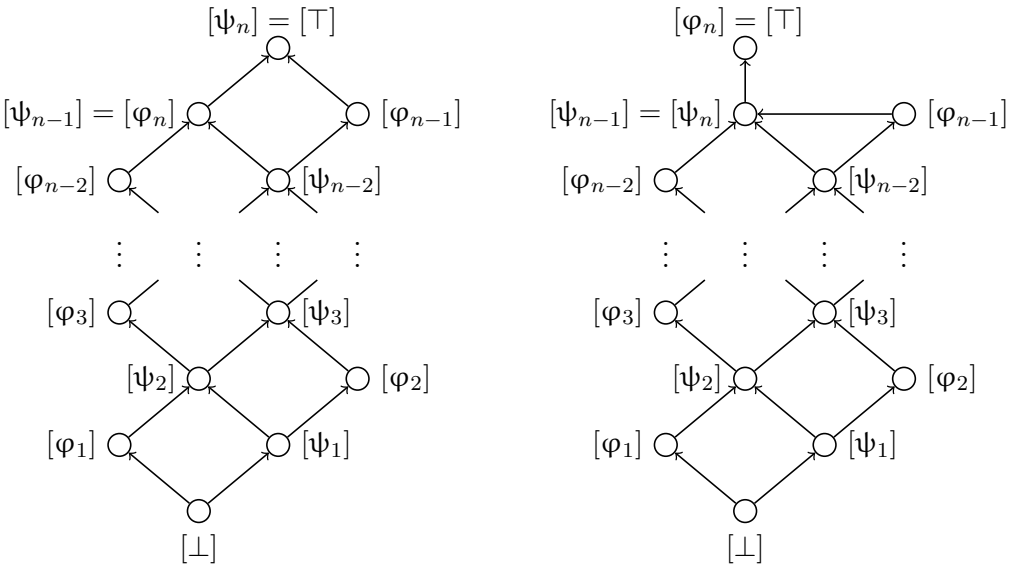

Abbildung 4.5: Auf der linken Seite ist der Verband von $\mathrm{IPL}[1]^{\psi_n}$ zu sehen und auf der rechten Seite ist derjenige von $\mathrm{IPL}[1]^{\varphi_n}$ dargestellt. Die Abbildung zeigt die Verbände für ungerade n.

Für die Logik KC[1] wollen wir die Formel- und Modellklassen hier noch einmal konkret angeben. Bei KC-Modellen ist die Sichtbarkeitsrelation eine gerichtete Halbordnung, jedes Modell hat also genau eine Welt ohne echte Nachfolger. Aus [DL59] folgt, dass diese Logik syntaktisch aus IPL durch die Hinzunahme des Axioms $\neg p \vee \neg\neg p$ entsteht. Es gilt $\neg p \vee \neg\neg p \equiv_{\mathfrak{K}^{\mathsf{i}}[1]} \psi_3$, also $\mathrm{KC}[1] = \mathrm{IPL}[1]^{\psi_3}$. Das bedeutet, jede Welt eines KC[1]-Modells hat den Modellindex $1, 2$ oder 3 und es gibt die sechs Formeläquivalenzklassen $[\bot], [\varphi_1], [\psi_1], [\varphi_2], [\psi_2]$ und $[\psi_3]$. In KC[1] sind ψ_2 und φ_3 äquivalent.
Die Aussagenlogik AL[1] mit einer Variablen ist gerade $\mathrm{IPL}[1]^{\psi_2}$. In dieser Logik haben Welten nur zwei möglich Modellindizes, entweder 1 – in der Welt ist p erfüllt – oder 2 – in der Welt ist p nicht erfüllt. Diese beiden Indizes entsprechen genau den beiden aussagenlogischen Belegungen, die p mit `true` oder mit `false` belegen. Die Formeläquivalenzklassen sind $[\bot], [p], [\neg p]$ und $[\top]$ und es gilt offensichtlich, dass $\psi_1 = p$ und $\neg p \twoheadrightarrow p = \varphi_2$ äquivalent sind. In der Aussagenlogik haben $\neg$ und $\neg$ bzw. $\rightarrow$ und $\twoheadrightarrow$ dieselbe Bedeutung.

4.5 Weitere Resultate

Wir betrachten jetzt das Formelauswertungsproblem für verschiedene Varianten von IPL[1]. Zuerst beschränken wir uns bei den Modellen auf Bäume. Da man statt der in Definition 4.4 eingeführten kanonischen Modelle auch spezielle Bäume verwenden kann, gibt es keine Änderungen bezüglich der Formeläquivalenzklassen. Trotzdem zeigen wir, dass für diese Variante von IPL[1] das Formelauswertungsproblem nicht mehr AC^1-hart ist (außer $\mathsf{LOGdetCFL}$ und AC^1 fallen zusammen). Im zweiten Teil schauen wir uns IPL[1] in der ursprünglichen Form an, verwenden hier aber eine andere Art, die Formeln zu kodieren. Sie werden als Graph und nicht als Zeichenkette dargestellt. Da in dieser Darstellung mehrfach vorkommende Teilformeln nur einmal kodiert werden, gilt Lemma 4.7 nicht mehr – die Länge der Rieger-Nishimura-Formeln wächst nicht mehr exponentiell. Wir zeigen, dass diese Variante von IPL[1] ein P-vollständiges Formelauswertungsproblem hat. Abschließend betrachten wir Formeln, in denen die Konstanten und atomaren Aussagen durch Rieger-Nishimura-Indizes ersetzt werden. Wir erweitern die Relation $\models$ für diese Formeln und zeigen, dass dann die Formelauswertung P-hart ist.

4.5.1 Baummodelle

Grundsätzlich lässt sich jedes intuitionistische Modell zu einem Baum „auffalten“, kann dabei aber sehr groß werden, da ein Modell mit n Welten bis zu 2^n viele Pfade enthält. Wir betrachten jetzt $\mathrm{IPL}[1]^B = (\mathfrak{F}^{\mathsf{i}}[1], \mathfrak{K}^{\mathsf{i}}[1]^B)$, wobei $\mathfrak{K}^{\mathsf{i}}[1]^B := \{\mathcal{M} \in \mathfrak{K}^{\mathsf{i}}[1] \mid \text{ der Rahmen von } \mathcal{M} \text{ ist ein Baum}\}$ ist – $\mathrm{IPL}[1]^B$ bildet also ein Fragment von IPL[1]. Für $\mathrm{IPL}[1]^B$ können wir die kanonische Baummodelle $\mathcal{H}_n^B$ $(n \geq 1)$ analog zu Definition 4.4 angeben. Statt einer formalen Definition zeigen wir deren Konstruktion in Abbildung 4.6.

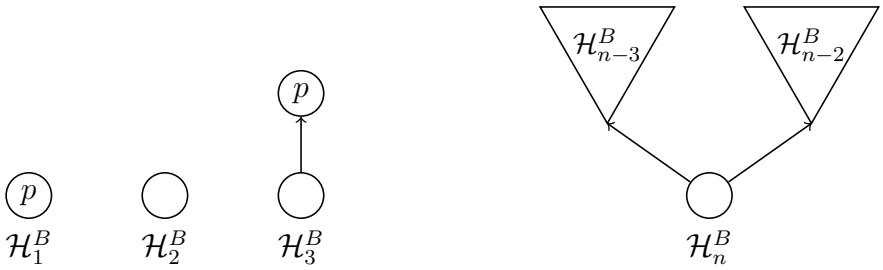

Abbildung 4.6: Die kanonischen Bäume $\mathcal{H}_1^B$, $\mathcal{H}_2^B$ und $\mathcal{H}_3^B$ auf der linken Seite als Anfang. Auf der rechten Seite ist zu sehen, wie der kanonische Baum $\mathcal{H}_n^B$ für $n > 3$ aufgebaut ist.

Die kanonischen Baummodelle entstehen aus den kanonischen Modellen, indem man sie auffaltet. Wir bezeichnen für $n \geq 1$ mit w_n die Wurzel von $\mathcal{H}_n^B$. Es ist nach Konstruktion offensichtlich, dass $(\mathcal{H}_n^B, w_n) \equiv_{\mathfrak{F}^{\mathrm{i}}[1]} (\mathcal{H}_n, n)$ für alle $n \geq 1$ gilt. Daraus folgt, dass die Formeläquivalenzklassen bezüglich $\equiv_{\mathfrak{K}^{\mathrm{i}}[1]}$ und bezüglich $\equiv_{\mathfrak{K}^{\mathrm{i}}[1]^B}$ dieselben sind. Ähnlich wie bei den Rieger-Nishimura-Formeln, in denen Teilformeln teilweise mehrfach vorkommen, kommen auch bei den Bäumen Welten mit einem Modellindex teilweise mehrfach vor. In den kanonischen Modellen gibt es exponentiell viele Pfade (in der Anzahl der Welten), aber es gibt keine zwei Welten mit demselben Modellindex. Da es in Bäumen einen linearen Zusammenhang zwischen der Anzahl der Knoten und der Anzahl der Pfade gibt, wachsen die Baummodelle exponentiell im Modellindex ihrer Wurzelwelt. Entsprechend gibt es in den kanonischen Baummodellen auch viele Welten, die denselben Modellindex haben. Für das Formelauswertungsproblem verschiebt sich damit die obere Schranke nach unten.

Theorem 4.25 *Das Problem* IPL[1]^B-FA *ist in* LOGdetCFL.

Beweis. Das Prinzip, nach dem eine Instanz entschieden wird, ist dem aus dem Beweis von Theorem 4.17 sehr ähnlich. Für die gegebene Formel wird der Rieger-Nishimura-Index bestimmt und für die Welt der Modellindex. Je nachdem, ob die Indizes passen oder nicht (siehe Bemerkung 4.14), wird akzeptiert oder abgelehnt. Der Rieger-Nishimura-Index kann mit den Ressourcen von LOGdetCFL bestimmt werden (Lemma 4.8). Die Berechnung des Modellindex in Algorithmus 6 hingegen benötigt die Ressourcen von AC^1 (siehe Beweis von Theorem 4.17). Da es sich bei den IPL[1]^B-Modellen ausschließlich um Bäume handelt, können wir hier einen Ansatz für die Berechnung der Modellindizes wählen, der nicht auf Alternierung basiert. Sei ModellIndex^B das Modellindexproblem ModellIndex eingeschränkt auf die IPL[1]^B-Modelle, dann gilt die folgende Behauptung.

Behauptung 4.4 *Das Problem* ModellIndex^B *ist in* LOGdetCFL.

Beweis der Behauptung. Algorithmus 7 entscheidet für ein Modell $\mathcal{M} \in \mathfrak{K}^{\mathsf{i}}[1]^B$, eine Welt w aus $\mathcal{M}$ und eine natürliche Zahl i, ob $\hbar(\mathcal{M}, w) = i$ gilt. Erst zeigen wir die Korrektheit, dann analysieren wir seine Komplexität und zeigen, dass die Ressourcen von LOGdetCFL ausreichen.

Die Funktion `Modellindex` berechnet rekursiv für eine Welt v aus einem Modell $\mathcal{N} = (W, \leqslant, \xi, \{p\})$ den Modellindex. Dabei durchläuft sie $\mathcal{N}$ nach dem Prinzip der Tiefensuche. In den Variablen $maxNF_1$ und $maxNF_2$ werden die beiden größten verschiedenen Modellindizes aller echten Nachfolger von v gespeichert, in $maxNF_1$ steht der größere Wert. Diese beiden Werte genügen nach Definition 4.11, um den Modellindex von v zu bestimmen. Außerdem ist es ausreichend, sich dabei ausschließlich die direkten Nachfolger[2] anzuschauen, da alle weiter entfernten Nachfolger auch Nachfolger von den direkten Nachfolgern sind. Hat v keine echten Nachfolger, dann hängt der Modellindex von der Belegungsfunktion ξ ab (Zeilen 3 bis 5). Gibt es echte Nachfolger, so werden alle direkten Nachfolger durchlaufen (Zeilen 7 bis 13). Für jeden direkten Nachfolger wird der Modellindex berechnet (Zeile 8). Wenn dieser größer als der größte, bisher berechnete Nachfolger-Modellindex ($maxNF_1$) ist, werden $maxNF_1$ und $maxNF_2$ entsprechend angepasst (Zeilen 9 bis 11). Liegt er zwischen den beiden größten, so nimmt $maxNF_2$ diesen Wert an und $maxNF_1$ bleibt unverändert (Zeilen 12 und 13). Wurden die Modellindizes von allen direkten Nachfolgern berechnet (die beiden sich unterscheidenden größten sind in $maxNF_1$ und $maxNF_2$ gespeichert), kann der Modellindex von v berechnet werden (Zeilen 14 bis 20). Gibt es nur echte Nachfolger mit Modellindex 1 (in allen ist p erfüllt), dann hängt der Modellindex von w von ξ ab (Zeilen 15 und 16). Haben hingegen alle echten Nachfolger den Modellindex 2 (p ist nicht erfüllt), so hat v selbst auch Modellindex 2 (Zeile 17). Wenn $maxNF_1$ und $maxNF_2$ positiv sind (Zeile 18) und sich um genau 1 unterscheiden (Zeile 19), dann bedeutet dies, dass v einen Nachfolger mit Modellindex $maxNF_2$ und einen mit $maxNF_1 = maxNF_2 + 1$ hat. Es gibt aber keinen echten Nachfolger, der einen größeren Modellindex hat. (Insbesondere gibt es keinen Nachfolger mit Modellindex $maxNF_1 + 1$.) Damit folgt aus Definition 4.11, dass w den Modellindex $maxNF_1 + 2$ hat. Unterscheiden sich $maxNF_1$ und $maxNF_2$ um mehr als 1, wird $maxNF_1$ zurückgegeben (Zeile 20). Für die Korrektheit betrachten wir 2 Fälle. Der Fall $maxNF_1 = 3$ ist nach Definition des Modellindexes (Definition 4.11) klar. Für den Fall $maxNF_1 > 3$ gilt, dass es einen echten Nachfolger v' von v mit $\hbar(\mathcal{N}, v') = maxNF_1$ gibt, also muss es echte Nachfolger $v_1, v_2 \in W$ von v' geben, für die $\hbar(\mathcal{N}, v_1) = maxNF_1 - 2$ und $\hbar(\mathcal{N}, v_2) = maxNF_1 - 3$ gilt. Da v keinen Nachfolger mit Modellindex $maxNF_1 - 1$ hat, aber sowohl v_1 als auch v_2 (wegen der Transitivität von $\leqslant$) sieht, gilt nach Definition 4.11 $\hbar(\mathcal{N}, v) = maxNF_1$. Damit ist gezeigt, dass Algorithmus 7 korrekt arbeitet.

Nun schauen wir uns die Komplexität an. Da es sich bei dem Rahmen von $\mathcal{M}$ um einen Baum handelt, ist jede Welt direkter Nachfolger von höchstens einer

[2] Eine Welt v' ist ein direkter Nachfolger von v, wenn es kein v'' mit $v \neq v'' \neq v'$ und $v \leqslant v'' \leqslant v'$ gibt.

Algorithmus 7 `Modellindexberechnung für Baummodelle`

Eingabe: Modell $\mathcal{M} \in \mathfrak{K}^{\mathsf{i}}[1]^B$, Welt $w \in W$, Zahl $i \in \mathbb{N}$.

1: **wenn** `Modellindex`$(\mathcal{M}, w) = i$ **dann** akzeptiere **sonst** lehne ab
2: **Funktion** `Modellindex`$(\mathcal{N} = (W, \leqslant, \xi, \{p\}), v)$ // gibt $\hbar(\mathcal{N}, v$ zurück
3: **wenn** w keine echten Nachfolger hat **dann**
4: **wenn** $v \in \xi(p)$ **dann Rückgabe** 1 **sonst Rückgabe** 2
5: $maxNF_1 := 0$
6: $maxNF_2 := 0$
7: **für** alle direkten Nachfolger v' von v **wiederhole**
8: $i :=$ `Modellindex`$(\mathcal{N}, v')$
9: **wenn** $i > maxNF_1$ **dann**
10: $maxNF_2 := maxNF_1$
11: $maxNF_1 := i$
12: **wenn** $maxNF_1 > i > maxNF_2$ **dann**
13: $maxNF_2 := i$
14: **wenn** $maxNF_2 = 0$ **dann**
15: **wenn** $maxNF_1 = 1$ und $w \in \xi(p)$ **dann Rückgabe** 1
16: **wenn** $maxNF_1 = 1$ und $w \notin \xi(p)$ **dann Rückgabe** 3
17: **wenn** $maxNF_1 = 2$ **dann Rückgabe** 2
18: **wenn** $maxNF_2 > 0$ **dann**
19: **wenn** $maxNF_1 = maxNF_2 + 1$ **dann Rückgabe** $maxNF_1 + 2$
20: **sonst Rückgabe** $maxNF_1$

Welt, wird also in der *für*-Schleife in Zeile 7 genau einmal behandelt. Die Funktion `Modellindex` wird demnach für jede Welt aus $\mathcal{M}$ genau einmal aufgerufen. Die Laufzeit ist somit polynomiell in der Eingabegröße.[3] Bei jedem Durchlauf müssen nur die Variablen $maxNF_1$ und $maxNF_2$ gespeichert werden. Da der Modellindex die Anzahl der Welten aus $\mathcal{M}$ nicht um mehr als 1 übersteigt, benötigen diese Variablen logarithmischen Speicherplatz. Die Rücksprungadressen für die Rekursion können auf einem Stapel gespeichert werden und benötigen ebenfalls logarithmischen Platz, da die Laufzeit polynomiell ist. Es folgt, dass die Funktion `Modellindex` mit LOGdetCFL-Ressourcen berechnet werden kann. ▪

Eine IPL[1]^B-FA-Instanz $\langle \alpha, \mathcal{M}, w \rangle$ prüfen wir wie folgt: Zuerst bestimmt man den Rieger-Nishimura-Index $RNIndex(\alpha)$ von α, dann den Modellindex $\hbar(\mathcal{M}, w)$ von w. Beides ist mit den Ressourcen von LOGdetCFL möglich (Lemma 4.8 und Behauptung 4.4). Die Entscheidung, ob die Instanz akzeptiert wird, ist gemäß Bemerkung 4.13 zu treffen. Damit folgt IPL[1]^B-FA $\in$ LOGdetCFL. □

Aus diesem Resultat kann man ableiten, dass für die AC^1-Härte von IPL[1]-FA die exponentielle Zahl der Pfade in einem IPL[1]-Modell offenbar wesentlich ist.

[3] In einem Baum kann für zwei Knoten v_1 und v_2 in logarithmischen Platz bestimmt werden, ob $v_1 \leqslant v_2$ oder $v_2 \leqslant v_1$ oder weder $v_1 \leqslant v_2$ noch $v_2 \leqslant v_1$ gilt.

4.5.2 Formeln als Graphen

Neben der Darstellung als Zeichenketten, lassen sich Formeln auch als Graphen darstellen. Operatoren, Variablen und Konstanten sind dabei die Knoten, die Struktur der Formel wird durch die Kanten bzw. die Pfade repräsentiert. Die Variablen und die Konstanten haben Ausgangsgrad 0 und n-stellige Operatoren haben Ausgangsgrad n. Jeder Knoten repräsentiert eine bestimmte (Teil-)Formel, die aus allen möglichen ausgehenden Pfaden besteht. Wir beschränken uns hier wieder auf die Formeln aus $\mathfrak{F}^{\mathsf{i}}[1]$. Die Knoten haben einen Ausgangsgrad von maximal 2. Da die Implikation $\rightarrowtriangle$ nicht kommutativ ist, werden die ausgehenden Kanten mit ℓ und r beschriftet, um eindeutig festzulegen, welche Teilformel auf der linken Seite und welche auf der rechten Seite von $\rightarrowtriangle$ steht. Ein Beispiel ist in Abbildung 4.7 zu sehen.

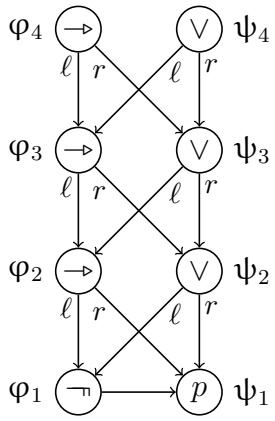

Abbildung 4.7: Darstellung der Rieger-Nishimura-Formeln φ_1 bis ψ_4 als Graph. Die jeweilige Formel startet in dem Knoten, der ihren Namen trägt und endet im Knoten p (bzw. ψ_1).

Die in Abbildung 4.7 gezeigten Rieger-Nishimura-Formeln verdeutlichen bereits, dass in dieser Art der Darstellung die Größe der Formeln nicht mehr exponentiell mit ihrem Rang wächst. In jeder Ebene kommen zwei Knoten und vier Kanten dazu. Das exponentielle Wachstum der Formeln in normaler Darstellung zeigt sich bei der Darstellung als Graph in der Anzahl der Pfade. Als Zeichenkette dargestellt, enthält ψ_4 die (Teil-)Formel φ_2 zweimal. In dem Graph, der φ_4 darstellt, gibt es zwei Pfade vom Knoten ψ_4 aus, die über den Knoten φ_2 führen. Diese beiden Pfade entsprechen dem zweimaligen Vorkommen von φ_2 als Teilformel in ψ_4. Damit verliert Lemma 4.7 für diese Logik seine Gültigkeit, der Speicherplatz, den eine Rieger-Nishimura-Formel benötigt, ist nur noch polynomiell in ihrem Rang. Die Bestimmung des Rieger-Nishimura-Index für eine Formel in Graphdarstellung, kann als Auswertung eines Schaltkreises polynomieller Größe gesehen werden. Die Auswertung solcher Schaltkreise ist P-hart [Lad75].

Theorem 4.26 *Wenn die* IPL[1]*-Formeln als Graphen dargestellt werden, dann sind für alle Rieger-Nishimura-Formeln* $\alpha \in \{\varphi_1, \psi_1, \varphi_2, \psi_2, \dots\}$ *folgende Probleme* P*-vollständig:*

(1) IPL[1]-FA *und* IPL[1]$^{\alpha}$-FA*,*

(2) IPL[1]-TAUT *und* IPL[1]$^{\alpha}$-TAUT *und*

(3) IPL[1]-SAT *und* IPL[1]$^{\alpha}$-SAT*.*

Beweis. P als obere Schranke ist für alle 3 Aussagen sofort klar. Die Auswertung von Schaltkreisen polynomieller Größe ist P-hart [Lad75]. Hat man einen beliebigen Schaltkreis und eine Belegung der Eingabegatter gegeben, so kann dieser Schaltkreis genau dann zu `true` ausgewertet werden, wenn die durch ihn repräsentierte aussagenlogische Formel wahr ist. Das entspricht der Frage, ob die Formel in einem Modell, bestehend aus nur einer, sich selbst sehenden Welt (unabhängig von der Belegung, da in dieser Formel keine Variablen vorkommen), erfüllt wird. Ein solches Modell kommt in jeder superintuitionistischen Variante von IPL[1] vor und damit folgt die P-Härte für alle 3 Aussagen sofort, da die Begriffe *gültig*, *erfüllbar* und *erfüllt* für Formeln ohne Variablen über den Modellen, bestehend aus nur einer Welt, zusammenfallen. □

Dieses Resultat zeigt eine weitere wesentliche Eigenschaft von IPL[1]-FA, denn für kompakte Darstellung der Formeln ist das Problem nicht mehr in AC^1 (außer im unwahrscheinlichen Fall, dass P und AC^1 zusammenfallen).
Aus der verwendeten Reduktion im obigen Beweis ergibt sich auch sofort folgende Bemerkung für intuitionistische Logiken mit Formeln ohne Variablen und Modelle ohne Reflexivität.

Bemerkung 4.27 *Wenn die Formeln als Graphen dargestellt werden, dann sind* L[0]-FA*,* L[0]-TAUT *und* L[0]-SAT *für* $\mathrm{L} \in \{\mathrm{BPL}, \mathrm{FPL}, \mathrm{IPL}, \mathrm{KC}, \mathrm{LC}, \mathrm{AL}\}$ P*-vollständig.*

Für die Logik FPL mit irreflexiven Modellen ergibt sich die P-Härte, da auch das Auswerten von Schaltkreisen ohne Negationsgatter P-hart ist und es für Formeln ohne Negation (und damit auch ohne Implikation) keine Rolle spielt, ob sich eine Welt selbst sieht oder nicht.

4.5.3 Formeln mit Rieger-Nishimura-Indizes

Die Formeln aus $\mathfrak{F}^{\mathrm{i}}[1]$ bestehen aus den Zeichen $\{p, \bot\} \cup \{\wedge, \vee, \rightarrowtriangle\}$ (zusätzlich kommen noch Klammern und möglicherweise Abkürzungen wie $\neg$ und $\top$ vor). Statt der Variablen p und den Konstanten $\bot$ und $\top$ kann man aber auch Rieger-Nishimura-Indizes verwenden. Dabei werden $p = \psi_1$ durch den Index $(1, psi)$ und die Konstanten $\bot$ und $\top$ durch ihre Indizes $(0, \bot)$ und $(0, \top)$ dargestellt. Die Formel $(p \rightarrowtriangle \bot) \vee \top$ ist dann $((1, psi) \rightarrowtriangle (0, \bot)) \vee (0, \top)$. Bis zu diesem Punkt

ändert sich für die Komplexität nichts. Man kann aber auch größere Teilformeln durch die Rieger-Nishimura-Indizes der zu ihnen äquivalenten Rieger-Nishimura-Formeln ersetzen. So wird zum Beispiel aus $((\neg p \rightarrowtriangle p) \vee (\neg p \vee p)) \wedge (\neg p \rightarrowtriangle p)$ die Formel $(3, psi) \wedge (2, phi)$. Wir bezeichnen die Menge aller Formeln aus $\mathfrak{F}^{\mathsf{i}}[1]$, bei denen jedes Vorkommen einer Rieger-Nishimura-Formel durch ihren Index ersetzt wurde, mit $\mathfrak{F}^{\mathsf{i}}[1]^{RNI}$.
Will man diese Formeln über Modellen aus $\mathfrak{K}^{\mathsf{i}}[1]$ interpretieren, muss man bei der induktiv definierten Relation $\models$ (Definition 2.5) den Induktionsanfang modifizieren. Seien (t, i) ein Rieger-Nishimura-Index, $\mathcal{M} \in \mathfrak{K}^{\mathsf{i}}[1]$ und w eine Welt aus $\mathcal{M}$. Dann gilt

$$\begin{aligned}
&\text{wenn } t = \bot \ : \ \mathcal{M}, w \not\models (0, t) \ , \\
&\text{wenn } t = \top \ : \ \mathcal{M}, w \models (0, t) \ , \\
&\text{wenn } t = psi \ : \ \mathcal{M}, w \models (t, i) \iff \hbar(\mathcal{M}, w) \leq i \quad \text{und} \\
&\text{wenn } t = phi \ : \ \mathcal{M}, w \models (t, i) \iff \hbar(\mathcal{M}, w) < i \text{ oder } \hbar(\mathcal{M}, w) = i + 1 \ .
\end{aligned}$$

Für die Operatoren bleibt $\models$ wie auch schon in Definition 2.5 erhalten. Mit dieser erweiterten Definition von $\models$ kann man jetzt auch das Formelauswertungsproblem für $\text{IPL}[1]^{RNI} = (\mathfrak{F}^{\mathsf{i}}[1]^{RNI}, \mathfrak{K}^{\mathsf{i}}[1])$ betrachten. (Die Modelle bleiben unverändert.)

Theorem 4.28 *Das Problem* $\text{IPL}[1]^{RNI}$-FA *ist* P*-vollständig.*

Beweis. Aus Theorem 2.30 folgt P auch als obere Schranke (Algorithmus 1 lässt sich einfach an die Formeln aus $\mathfrak{F}^{\mathsf{i}}[1]^{RNI}$ anpassen).
Für die untere Schranke geben wir eine Reduktion von Asgep auf das Formelauswertungsproblem an. Sei $\langle G, s, t\rangle$ eine Asgep-Instanz, wobei G aus m Schichten besteht. Weiter sei $\mathcal{M}_G$ das Modell, das aus $\langle G, s, t\rangle$ gemäß der Konstruktion aus Abschnitt 4.3.1 entsteht ($\mathcal{M}_G = g(\langle G, s, t\rangle)$). Aus Lemma 4.20 und Bemerkung 4.14 wissen wir, dass

$$\langle G, s, t\rangle \in \text{Asgep} \iff \hbar(\mathcal{M}_G, s^{out}) = 4m - 2 \iff \mathcal{M}_G, s^{out} \models \varphi_{4m-3}$$

gilt. In der Sprache von $\mathfrak{F}^{\mathsf{i}}[1]^{RNI}$ heißt dies

$$\langle G, s, t\rangle \in \text{Asgep} \iff \mathcal{M}_G, s^{out} \models (phi, 4m - 3) \ .$$

Damit folgt die P-Härte für $\text{IPL}[1]^{RNI}$-FA aus Theorem 2.21. □

Der Beweis verläuft analog zum Beweis von Theorem 4.22 (Beweis der AC^1-Härte von IPL[1]-FA), der Unterschied besteht lediglich darin, dass wir hier Asgep-Instanzen statt $\text{Asgep}_{\log}$-Instanzen verwenden können, da die Formeln aus $\mathfrak{F}^{\mathsf{i}}[1]^{RNI}$ kurz sind. Wesentlich dafür ist, dass der Rieger-Nishimura-Index $(phi, 4m - 3)$ nicht exponentiell groß in m ist. Auch diese Art IPL[1]-Formeln kompakt darzustellen, führt dazu, dass das Formelauswertungsproblem nicht mehr in AC^1 ist (außer P und AC^1 fallen zusammen).

4.6 Zusammenfassung

Das Kapitel 4 widmete sich im Wesentlichen der intuitionistischen Logik mit einer Variablen. Wir haben IPL[1] in der ursprünglichen Form, die superintuitionistischen Fragmente $\mathrm{IPL}[1]^{\alpha}$ und einige speziellere Varianten untersucht. Der Fokus lag dabei auf dem Formelauswertungsproblem, wir konnten aber auch Ergebnisse für das Tautologie-, das Erfüllbarkeits- und das Modelläquivalenzproblem angeben. Eine Übersicht über die Ergebnisse liefert Tabelle 4.3.

Problem \ Logik	IPL[1]	$\mathrm{IPL}[1]^{\alpha}$	$\mathrm{IPL}[1]^{B}$	$\mathrm{IPL}[1]^{G}$	$\mathrm{IPL}[1]^{RNI}$
FA	AC^1	NC^1	$\in \mathsf{LOGdetCFL}$	P	P
SAT	NC^1	NC^1	NC^1	P	$\in \mathsf{LOGdetCFL}$
TAUT	$\in \mathsf{LOGdetCFL}$	NC^1	$\in \mathsf{LOGdetCFL}$	P	$\in \mathsf{LOGdetCFL}$
MÄQ	P	$\in \mathsf{AC}^0$	$\in \mathsf{LOGdetCFL}$	P	P

Tabelle 4.3: Zusammenfassung der Ergebnisse aus Kapitel 4. Die Logiken $\mathrm{IPL}[1]^{B}$, $\mathrm{IPL}[1]^{G}$ und $\mathrm{IPL}[1]^{RNI}$ sind die in Abschnitt 4.5 betrachteten Varianten von IPL[1]. Bei $\mathrm{IPL}[1]^{B}$ werden nur Bäume als Modelle verwendet, bei $\mathrm{IPL}[1]^{G}$ sind die Formeln als Graphen kodiert und bei $\mathrm{IPL}[1]^{RNI}$ kommen in den Formeln statt den Konstanten und der Variablen p Rieger-Nishimura-Indizes vor. Für alle $\mathsf{LOGdetCFL}$-Resultate ist NC^1 die untere Schranke. Alle Ergebnisse ohne „$\in$“ sind Vollständigkeitsresultate.

Unser Hauptresultat bezieht sich auf die Komplexität der Formelauswertung in IPL[1].

Theorem 4.29 *Das Problem* IPL[1]-FA *ist* AC^1*-vollständig.*

Die obere Schranke liefert das Theorem 4.17, der zentrale Punkt dabei ist das exponentielle Längenwachstum der IPL[1]-Formeln. Aus Theorem 4.22 kommt die untere Schranke, hier verwenden wir eine Reduktion von $\textsc{Asgep}_{\log}$ auf IPL[1]-FA. Bemerkenswert an diesem Resultat ist, dass dieses Problem nach unserer Kenntnis das erste ist, bei dem die spezielle Eigenschaft von AC^1 – die logarithmische Beschränkung der Alternierungszahl – kein direkter Bestandteil der Problemdefinition ist. Bei der Analyse des Formelauswertungsproblem erkennt man diese Beschränkung erst bei der genauen Betrachtung der Formeln und ihrer Äquivalenzklassen. Anschaulich gesprochen wächst die Länge der Formeln von Äquivalenzklasse zu Äquivalenzklasse exponentiell. Aus diesem exponentiellen Wachstum können wir für die Berechnung schließen, dass die Anzahl der notwendigen Alternierungen im Verhältnis zur Größe der Eingabe logarithmisch ist. Die Zahl der

Alternierungen entspricht wiederum genau der Zahl der Schichten von alternierenden Graphen und ermöglicht so eine Reduktion von $\text{ASGEP}_{\log}$ auf IPL[1]-FA. Aus den Bemerkungen 4.16 und 4.21 folgt, dass für IPL[1] das Modelläquivalenzproblem IPL[1]-MÄQ P-vollständig ist. Das Erfüllbarkeitsproblem IPL[1]-SAT ist NC^1-vollständig, die obere Schranke liefert Bemerkung 4.10. Eine IPL[1]-Formel ist genau dann erfüllbar, wenn es ein Modell bestehend aus nur einer Welt, die sich selbst sieht, gibt, in dem sie erfüllt wird. Die untere Schranke folgt, da bereits die Formelauswertung in der Aussagenlogik ohne Variablen NC^1-hart ist (Theorem 2.26 bzw. [Bus87]). Für das Tautologieproblem IPL[1]-TAUT konnten wir zeigen, dass es in $\mathsf{LOGdetCFL}$ liegt (Bemerkung 4.9). Da eine Formel ohne Variablen (nur mit Konstanten) genau dann eine Tautologie in IPL[1] ist, wenn sie aussagenlogisch gültig ist (also eine Ja-Instanz von $\text{BFVP}[\wedge, \vee, \neg]$ ist), folgt für IPL[1]-TAUT auch die NC^1-Härte. Die exakte Komplexität ist noch offen.

In Abschnitt 4.4 betrachteten wir superintuitionistische Logiken mit einer Variablen. Für das Formelauswertungsproblem konnten wir folgendes Resultat zeigen.

Theorem 4.30 *Für jede Rieger-Nishimura-Formel* $\alpha \in \{\varphi_1, \psi_1, \varphi_2, \psi_2, \dots\}$ *ist das Problem* $\text{IPL}[1]^\alpha$-FA NC^1*-vollständig.*

Die obere Schranke (Theorem 4.23) basiert darauf, dass die Logiken alle endlich erzeugt sind. Die NC^1-Härte (Theorem 4.24) ist letztendlich wieder eine Folgerung aus Theorem 2.26 bzw. [Bus87].
Das Erfüllbarkeitsproblem für diese Logiken ist ebenfalls NC^1-vollständig. Die obere Schranke folgt aus Theorem 3.4 und die untere Schranke analog zum allgemeinen Fall IPL[1]-SAT aus Theorem 2.26 bzw. [Bus87]. Auch für das Tautologieproblem folgt NC^1 als obere Schranke aus Theorem 3.4, NC^1 als untere Schranke gilt wieder analog zum allgemeinen Fall. Nach Lemma 3.5 gilt $\text{IPL}[1]^\alpha$-MÄQ $\in \mathsf{AC}^0$ für $\alpha \in \{\varphi_1, \psi_1, \varphi_2, \psi_2, \dots\}$.

In Abschnitt 4.5 untersuchten wir verschiedene Varianten von IPL[1]. Dieser Abschnitt grenzt das Resultat aus Theorem 4.29 weiter ab.
Lässt man als Modelle nur Bäume zu, so ist das Formelauswertungsproblem leichter als AC^1. Konkret liefert Theorem 4.25, dass es in $\mathsf{LOGdetCFL}$ liegt. Die Komplexität bei Tautologie- und Erfüllbarkeitsproblem bleibt unverändert gegenüber IPL[1]. Für das Modelläquivalenzproblem ergibt sich hier ebenfalls $\mathsf{LOGdetCFL}$ als obere Schranke (folgt aus Behauptung 4.4).
Kodiert man hingegen Formeln als Graphen, so fällt die Eigenschaft des exponentiellen Längenwachstums weg. Damit werden das Formelauswertungs-, das Tautologie- und das Erfüllbarkeitsproblem P-vollständig (Theorem 4.26). Interessanterweise überträgt sich das auch auf die superintuitionistischen Logiken. Das Modelläquivalenzproblem bleibt unverändert gegenüber IPL[1] P-vollständig.
Abschließend betrachteten wir noch eine sehr abstrakte Version, bei der in Formeln nicht nur Variablen und Konstanten vorkommen, sondern auch Rieger-Nishimura-Indizes verwendet werden dürfen. Auch in diesem Fall ergibt sich für das Formelauswertungsproblem die P-Vollständigkeit (Theorem 4.28), da diese Modifikation

wieder zu kurzen Formeln führt. Der Rieger-Nishimura-Index einer solchen Formel lässt sich mit Rekursion durch die Formel berechnen (analog zu Algorithmus 5). Damit folgt, dass sowohl Tautologie- als auch Erfüllbarkeitsproblem in $\mathsf{LOGdetCFL}$ liegen, das Modelläquivalenzproblem bleibt wieder unverändert gegenüber IPL[1] P-vollständig.
Diese drei Resultate zeigen, welche Aspekte für die AC^1-Vollständigkeit des Problems IPL[1]-FA wesentlich sind. Auf der einen Seite ist es wichtig, dass es in den Modellen exponentiell viele Pfade gibt, ist dies nicht gegeben, so wird die Formelauswertung leichter (Abschnitt 4.5.1). Andererseits dürfen die Formeln auch nicht kompakt kodiert werden, dies macht die Formelauswertung schwerer (Abschnitte 4.5.2 und 4.5.3).

Kapitel 5

Die P-harten Fälle

In diesem Kapitel beschäftigen wir uns mit intuitionistischen Logiken, deren Formelauswertungsproblem P-hart ist. Dabei geht es einerseits darum, die P-Härte zu zeigen (Abschnitt 5.1). Andererseits nehmen eine Abgrenzung hinsichtlich der Zahl der Variablen vor und zeigen, dass es für die von uns betrachteten Fragmente nicht möglich ist, weitere Variablen zu sparen, ohne die P-Härte zu verlieren. (Abschnitt 5.2). Bekannt ist bereits, dass das Formelauswertungsproblem für alle Logiken aus Definition 2.6 in P (Theorem 2.30) liegt. Die Ergebnisse dieses Kapitels werden in Abschnitt 5.3 nochmals zusammengefasst.

5.1 Fragmente von BPL, FPL, IPL und KC

Wir zeigen in diesem Abschnitt für bestimmte Fragmente von BPL, FPL, IPL und KC, dass sie ein P-hartes Formelauswertungsproblem haben.
Im Abschnitt 5.1.1 betrachten wir intuitionistische Logiken, in denen nur die intuitionistische Implikation $\rightarrow$ als Operator vorkommt – sogenannte Implikationsfragmente. Wir zeigen, dass das Implikationsfragment von KC ein P-hartes Formelauswertungsproblem hat. Dieses Resultat lässt sich direkt auf die Implikationsfragmente von IPL und BPL übertragen. In Abschnitt 5.1.2 beschränken wir die Zahl der Variablen und zeigen, dass das Formelauswertungsproblem für die Implikationsfragmente von FPL und BPL mit nur einer Variablen P-hart ist. In Abschnitt 5.1.3 lassen wir wieder alle Operatoren zu und zeigen P-Härte für die Formelauswertung in BPL ganz ohne Variablen und in KC bzw. IPL mit 2 Variablen.
Die P-Härte wird in diesem Abschnitt oft mit einer Reduktion von dem P-harten Problem ASGEP gezeigt (Theorem 2.21). Die alternierenden Schichtgraphen sind dabei die Basis für die Rahmen intuitionistischer Modelle. Da diese Graphen nicht transitiv sind und man in logarithmischem Platz nicht die transitive Hülle eines Graphen bestimmen kann, verwenden wir wieder (wie auch schon in Abschnitt 4.3) die pseudotransitive Hülle (Definition 2.23). Die so entstehenden Rahmen versehen wir mit einer Belegungsfunktion und einigen Logik-spezifischen Modifikationen und geben dann eine Formel an, die in einer bestimmten Welt des Modells ausgewertet wird.

5.1.1 Das Implikationsfragment von KC

Wir nennen eine intuitionistische Logik ein *Implikationsfragment*, wenn $\rightarrowtriangle$ der einzige Operator in den Formeln ist, KC[$\rightarrowtriangle$] bezeichnet also die Logik $(\mathfrak{F}^{\mathsf{i}}[\rightarrowtriangle], \mathfrak{K}^{\mathsf{i}}_{gerHO})$. Sollte das Fragment keine Variablen enthalten, so lassen wir zusätzlich noch die Konstante $\bot$ zu, da das Fragment sonst überhaupt keine Formeln enthalten würde. Lässt man die Konstante $\bot$ zu, so kann man in den Implikationsfragmenten auch die intuitionistische Negation darstellen.
Zuerst betrachten wir das Implikationsfragment von KC. Implikationsfragmente von KC und von IPL mit einer beschränkten Variablenzahl sind endlich erzeugt (Theorem 3.7), daher ist ihr Formelauswertungsproblem in NC^1 (Theorem 3.3). Außer im unwahrscheinlichen Fall, dass NC^1 und P zusammenfallen, ist es für die P-Härte der Formelauswertung in einem Implikationsfragment von KC notwendig, dass die Zahl der Variablen unbeschränkt ist. Weitere Operatoren oder die Konstante $\bot$ benötigen wir nicht.

Theorem 5.1 *Das Problem* KC[$\rightarrowtriangle$]-FA *ist* P*-hart.*

Beweis. Wir zeigen Asgep $\leq^{\log}_{\mathrm{m}}$ KC[$\rightarrowtriangle$]-FA, dann folgt mit Theorem 2.21, dass KC[$\rightarrowtriangle$]-FA P-hart ist.
Sei $\langle G, s, t\rangle$ eine Instanz von Asgep. Wir konstruieren ein KC[$\rightarrowtriangle$]-Modell $\mathcal{M}_G = (W, \leqslant, \xi, \mathrm{VAR})$ und eine KC[$\rightarrowtriangle$]-Formel α_G so, dass

$$\langle G, s, t\rangle \in \textsc{Asgep} \iff \mathcal{M}_G, s \models \alpha_G$$

gilt. Dabei ist $G = (V, E)$ ein alternierender Schichtgraph mit m Schichten und $V = V_\exists \cup V_\forall$. O.B.d.A. ist m gerade und $V_\exists = V_1 \cup V_3 \cup \cdots \cup V_{m-1}$ und $V_\forall = V_2 \cup V_4 \cup \cdots \cup V_m$. Wir setzen $V_{\geq i} := \bigcup_{m \geq j \geq i} V_j$.
Um aus G das Modell $\mathcal{M}_G$ zu konstruieren, fügen wir eine zusätzliche Welt hinzu, die von allen anderen gesehen wird. Wir bilden als nächstes die reflexive und pseudotransitive Hülle. Abschließend geben wir eine Belegungsfunktion an, mit der wir die verschiedenen Schichten voneinander unterscheiden können und die außerdem die Zielwelt t auszeichnet.
Wir setzen

$$\begin{aligned} V_{m+1} &:= \{top\}\ , \\ W &:= V_1 \cup V_2 \cup \ldots V_{m+1}\ . \end{aligned}$$

Weiter ist

$$E' := E \cup \{(v, top) \mid v \in V\}\ .$$

Für die Sichtbarkeitsrelation ergänzen wir die reflexiven und pseudotransitiven Kanten (Definition 2.23):

$$\begin{aligned} E^{refl} &:= W \times W\ , \\ E^{trans} &:= \bigcup_{i=1}^{m-1} V_i \times V_{\geq i+2}\ . \end{aligned}$$

Damit ergibt sich die Sichtbarkeitsrelation $\leqslant$ von $\mathcal{M}_G$ wie folgt:

$$\leqslant \;:=\; E' \cup E^{refl} \cup E^{trans} \;.$$

Es ist klar, dass $(W, \leqslant)$ aus G in logarithmischem Platz berechnet werden kann. Wir verwenden die Variablen $p_1, p_2, \ldots, p_{m+1}$. Jede Variable p_i $(1 \leq i \leq m)$ ist in den Welten der Schichten $V_{i+1}, V_{i+2}, \ldots, V_{m+1}$ erfüllt, p_m ist zusätzlich in der Zielwelt t erfüllt und p_{m+1} in *top*. Die Sichtbarkeitsrelation ξ ist wie folgt definiert:

$$\begin{aligned}
\xi(p_i) &:= V_{i+1} \cup V_{i+2} \cup \cdots \cup V_{m+1} \quad \text{für } i = 1, 2, \ldots, m-1 \;, \\
\xi(p_m) &:= \{t, \mathit{top}\} \;, \\
\xi(p_{m+1}) &:= \{\mathit{top}\} \;, \\
\xi(q) &:= \varnothing \quad \text{für } q \notin \{p_1, p_2, \ldots, p_{m+1}\} \;.
\end{aligned}$$

Damit ist

$$\mathcal{M}_G \;:=\; (W, \leqslant, \xi, \mathrm{VAR})$$

ein KC[$\rightarrowtriangle$]-Modell, das aus G in logarithmischem Platz berechnet werden kann. Abbildung 5.1 zeigt einen alternierenden Schichtgraphen G mit $m = 4$ Schichten und das daraus konstruierte Modell $\mathcal{M}_G$.

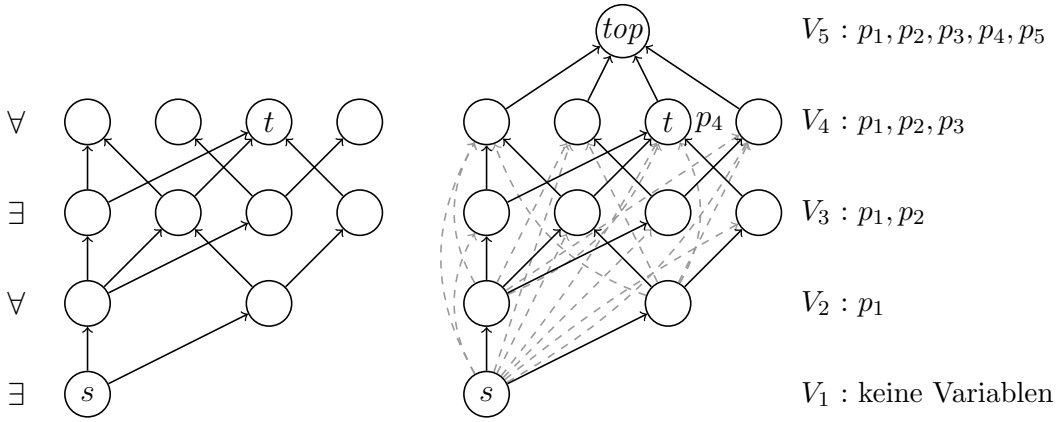

Abbildung 5.1: Ein alternierender Schichtgraph G (links) und das transformierte KC[$\rightarrowtriangle$]-Modell $\mathcal{M}_G$ (rechts) nach Konstruktion aus dem Beweis von Theorem 5.1. Die transitiven Kanten zum Knoten *top* und die reflexiven Kanten sind nicht abgebildet, die pseudotransitiven Kanten bis in die vorletzte Schicht sind grau gestrichelt dargestellt. Die Belegungen der Welten in den Schichten stehen rechts neben den Schichten, zusätzlich ist p_4 noch in der Welt t erfüllt.

Wir geben jetzt Formeln $\alpha_1, \alpha_2, \ldots, \alpha_m$ an, mit denen wir die $aPfad_G$-Eigenschaft von G in $\mathcal{M}_G$ ausdrücken wollen:

$$\begin{aligned} \alpha_m &:= p_m \twoheadrightarrow p_{m+1} \ , \\ \alpha_i &:= \alpha_{i+1} \twoheadrightarrow p_i \quad \text{für } i = m-1, m-2, \ldots, 1 \ . \end{aligned}$$

Ob die Formeln α_i erfüllt sind oder nicht, hängt nur von ursprünglichen Kanten aus G ab. Die pseudotransitiven und die reflexiven Kanten, die wir ergänzt haben, spielen dabei keine Rolle.

Behauptung 5.1 *Für $i = 1, 2, \ldots, m-1$ sind folgende Eigenschaften erfüllt:*

(1) $\forall w \in V_{\geq i+1} : \mathcal{M}_G, w \models \alpha_i$.

(2) $\forall w \in V_i : \mathcal{M}_G, w \models \alpha_i \iff \mathcal{M}_G, w \not\models \alpha_{i+1}$.

(3) $\forall w \in V_i : \mathcal{M}_G, w \models \alpha_i \iff \exists v \in V_{i+1}, w \leqslant v : \mathcal{M}_G, v \not\models \alpha_{i+1}$.

Beweis der Behauptung. Es sei $i \in \{1, 2, \ldots, m-1\}$.
Für (1) betrachten wir die Formel

$$\alpha_i = (\ldots((p_m \twoheadrightarrow p_{m+1}) \twoheadrightarrow p_{m-1}) \twoheadrightarrow \ldots \twoheadrightarrow p_{i+1}) \twoheadrightarrow p_i \ .$$

Sie ist in allen Welten aus $V_{\geq i+1}$ erfüllt, da $\xi(p_i) = V_{\geq i+1}$ gilt und somit die rechte Seite der Implikation in allen Welten aus $V_{\geq i+1}$ erfüllt ist.
Die Aussage (2) drückt aus, dass sich α_i und α_{i+1} in Schicht V_i genau gegensätzlich verhalten. Dies zeigen wir für $w \in V_i$ mit den folgenden Äquivalenzen:

$$\begin{aligned} &\mathcal{M}_G, w \models \alpha_i && \text{(i)} \\ \Leftrightarrow\ &\forall v, w \leqslant v : \mathcal{M}_G, v \models \alpha_{i+1} \Rightarrow \mathcal{M}_G, v \models p_i && \text{(ii)} \\ \Leftrightarrow\ &\mathcal{M}_G, w \models \alpha_{i+1} \Rightarrow \mathcal{M}_G, w \models p_i && \text{(iii)} \\ \Leftrightarrow\ &\mathcal{M}_G, w \not\models \alpha_{i+1} \ . && \text{(iv)} \end{aligned}$$

Die Äquivalenz zwischen (i) und (ii) folgt aus der Definition von $\twoheadrightarrow$. Aus $\xi(p_i) = V_{\geq i+1}$ folgt die Äquivalenz zwischen (ii) und (iii). Weil $\mathcal{M}_G, w \not\models p_i$ nach Konstruktion gilt, sind auch (iii) und (iv) äquivalent.
Bei Aussage (3) betrachten wir für ein $w \in V_i$ die beiden Richtungen getrennt.
Für „$\Leftarrow$“ sei $v \in V_{i+1}$ mit $w \leqslant v$ und $\mathcal{M}_G, v \not\models \alpha_{i+1}$. Wegen der Monotonie gilt dann auch $\mathcal{M}_G, w \not\models \alpha_{i+1}$ und mit Aussage (2) folgt $\mathcal{M}_G, w \models \alpha_i$.
Für „$\Rightarrow$“ gelte $\mathcal{M}_G, w \models \alpha_i$. Wir treffen folgende Annahme:

$$\forall v \in V_{i+1}, w \leqslant v : \mathcal{M}_G, v \models \alpha_{i+1} \ . \quad \text{(i)}$$

Damit ergeben sich folgende Konsequenzen:

$$\begin{aligned}
&\forall v \in V_{i+1}, w \leqslant v : \mathcal{M}_G, v \not\models \alpha_{i+2} && \text{(ii)}\\
\Rightarrow\quad & \mathcal{M}_G, w \not\models \alpha_{i+2} && \text{(iii)}\\
\Rightarrow\quad & \forall u \in W, w \leqslant u : \mathcal{M}_G, u \models \alpha_{i+2} \Rightarrow \mathcal{M}_G, u \models p_{i+1} && \text{(iv)}\\
\Rightarrow\quad & \mathcal{M}_G, w \models \alpha_{i+1} && \text{(v)}\\
\Rightarrow\quad & \mathcal{M}_G, w \not\models \alpha_i\ . && \text{(vi)}
\end{aligned}$$

Dies ist ein Widerspruch zur Voraussetzung, also ist die Annahme falsch und (3) bewiesen. Die Implikationen lassen sich wie folgt begründen: Aus (i) folgt (ii), da p_{i+1} in keiner Welt aus V_{i+1} erfüllt ist und $\alpha_{i+1} = \alpha_{i+2} \rightarrowtriangle p_{i+1}$ gilt. Aus (ii) folgt wegen der Monotonie (iii). In Welt w und ihren Nachfolgern aus Schicht V_{i+1} ist α_{i+2} nicht erfüllt, in allen weiteren Nachfolgen von w (aus $V_{\geq i+2}$) ist p_{i+1} nach Konstruktion erfüllt. Deswegen folgt (iv) aus (ii) und (iii). Aus (iv) folgt (v) wegen der Interpretation von $\rightarrowtriangle$. Die letzte Folgerung gilt schließlich wegen (2). ▪

Wir wollen nun zeigen, dass α_1 in der Welt $s \in V_1$ genau dann erfüllt ist, wenn es in G einen alternierenden Pfad von s nach t gibt – also $aPfad_G(s,t)$ gilt. Dafür geben wir eine weitere Vorbetrachtung an.

Behauptung 5.2 *Für alle $i = m, m-1, \ldots, 1$ und alle Welten $w \in V_i$ gilt*

$$aPfad_G(w,t) \iff \begin{cases} \mathcal{M}_G, w \models \alpha_i, & \textit{falls} \quad i \textit{ ungerade} \\ \mathcal{M}_G, w \not\models \alpha_i, & \textit{falls} \quad i \textit{ gerade}\ . \end{cases}$$

Beweis der Behauptung. Wir zeigen diese Behauptung mit Induktion über i. Im Induktionsanfang $i = m$ ist i eine gerade Zahl. Für $w \in V_m$ sind die folgenden Äquivalenzen offensichtlich:

$$\begin{aligned}
&aPfad_G(w,t)\\
\Leftrightarrow\quad & w = t\\
\Leftrightarrow\quad & \mathcal{M}_G, w \not\models p_m \rightarrowtriangle p_{m+1} \quad (= \alpha_m)\ .
\end{aligned}$$

Im Induktionsschritt ist $i < m$. Wir beweisen die Fälle i gerade und i ungerade getrennt. Zuerst gehen wir davon aus, dass i ungerade und V_i somit eine $\exists$-Schicht ist. Für $w \in V_i$ gelten die folgenden Äquivalenzen:

$$\begin{aligned}
&aPfad_G(w,t) && \text{(i)}\\
\Leftrightarrow\quad & \exists u, (w,u) \in E : aPfad_G(u,t) && \text{(ii)}\\
\Leftrightarrow\quad & \exists u \in V_{i+1}, w \leqslant u, : \mathcal{M}_G, u \not\models \alpha_{i+1} && \text{(iii)}\\
\Leftrightarrow\quad & \mathcal{M}_G, w \models \alpha_i\ . && \text{(iv)}
\end{aligned}$$

Die Äquivalenz zwischen (i) und (ii) folgt direkt aus der Definition der *aPfad*-Eigenschaft (Definition 2.18). Aus der Induktionsvoraussetzung und der Konstruk-

tion von $\mathcal{M}_G$ folgt, dass (ii) und (iii) äquivalent sind. Behauptung 5.1(3) liefert die Äquivalenz zwischen (iii) und (iv).
Jetzt gehen wir von einem geraden i aus. Die Schicht V_i ist also eine $\forall$-Schicht und für $w \in V_i$ gelten folgende Äquivalenzen:

$$\begin{aligned} & aPfad_G w, t && \text{(i)} \\ \Leftrightarrow \quad & \forall u, (w,u) \in E : aPfad_G(u,t) && \text{(ii)} \\ \Leftrightarrow \quad & \forall u \in V_{i+1}, w \leqslant u : \mathcal{M}_G, u \models \alpha_{i+1} && \text{(iii)} \\ \Leftrightarrow \quad & \mathcal{M}_G, w \not\models \alpha_i \ . && \text{(iv)} \end{aligned}$$

Wieder sind (i) und (ii) wegen der Definition der *aPfad*-Eigenschaft äquivalent. Die Äquivalenz zwischen (ii) und (iii) folgt aus der Induktionsvoraussetzung zusammen mit der Konstruktion von $\mathcal{M}_G$ und (iii) und (iv) sind wegen Behauptung 5.1(3) äquivalent. ▪
Wie setzen jetzt

$$\alpha_G \quad := \quad \alpha_1$$

und aus Behauptung 5.2 folgt sofort

$$\langle G, s, t\rangle \in \textsc{Asgep} \quad \Longleftrightarrow \quad \mathcal{M}_G, s \models \alpha_G \ .$$

Da sich $\mathcal{M}_G$ und α_G in logarithmischem Platz konstruieren lassen, gilt $\textsc{Asgep} \leq_{\mathrm{m}}^{\log}$ KC[$\rightarrow$] und mit Theorem 2.21 folgt, dass KC[$\rightarrow$]-FA P-hart ist. □

Theorem 5.1 lässt sich direkt auf die Implikationsfragmente BPL[$\rightarrow$] und IPL[$\rightarrow$] übertragen, da jedes KC[$\rightarrow$]-Modell auch eines für BPL[$\rightarrow$] bzw. IPL[$\rightarrow$] ist.

Bemerkung 5.2 *Die Probleme* IPL[$\rightarrow$]-FA *und* BPL[$\rightarrow$]-FA *sind* P*-hart.*

Die Idee der Konstruktion aus dem Beweis von Theorem 5.1 benutzen wir auch in den folgenden Abschnitten immer wieder. Der Rahmen des Modells $\mathcal{M}_G$ enthält immer noch die wesentlichen Merkmale des alternierenden Schichtgraphen G. Die Struktur der Schichten ist aber durch die zusätzlichen Kanten für Reflexivität und Transitivität schwer zu erkennen. Um die Information über die Zugehörigkeit einer Welt zu einer Schicht zu erhalten, verwenden wir die Belegungsfunktion. Diese gibt uns eine Art Distanzmaß, mit dem wir prüfen, wie weit ein Knoten von der obersten Schicht entfernt liegt. Außerdem wird die Zielwelt t von den anderen Welten der obersten Schicht unterscheidbar gemacht. Mit der Formel α_G nutzen wir diese Eigenschaften, um zu prüfen, ob es in der Struktur des ursprünglichen Graphen G einen alternierenden Pfad von s zu t gibt.

5.1.2 Das Implikationsfragment von FPL mit einer Variablen

In diesem Abschnitt schauen wir uns im Wesentlichen das Implikationsfragment von FPL mit nur einer Variablen an. Im Beweis der P-Härte des Formelaus-

wertungsproblems verwenden wir diesmal eine Reduktion vom Komplement von ASGEP. Die Grundidee ist aber ähnlich zu der aus dem Beweis von Theorem 5.1. Wir konstruieren erneut aus einem alternierenden Schichtgraphen ein Modell und eine Formel. Beschränkt man jedoch die Zahl der Variablen, so kann man die Belegungsfunktion nicht mehr so einfach als Maß für die Entfernung zur obersten Schicht verwenden. Wir benutzen jetzt die Irreflexivität der FPL-Modelle, um die Distanz einer Welt zur obersten Schicht zu bestimmen. Wesentlich ist dabei die Eigenschaft, dass Implikationen nur noch in echten Nachfolgern einer Welt (und nicht mehr in der Welt selbst) ausgewertet werden. Zum Beispiel ist die Formel $\top \mathrel{-\!\!\triangleright} \bot$ in einer Welt, die keinen Nachfolger hat erfüllt und $(\top \mathrel{-\!\!\triangleright} \bot) \mathrel{-\!\!\triangleright} \bot$ ist in einer Welt erfüllt, wenn keiner ihrer Nachfolger selbst noch einen Nachfolger hat. An diesem kleinen Beispiel kann man schon die neue Idee der Entfernungsmessung erkennen. Nur für die Unterscheidung der Zielwelt von den anderen Welten in der obersten Schicht benötigen wir noch eine Variable. Die Konstante $\bot$ verwenden wir für die Entfernungsmessung und um die Variable negieren zu können. Später in Theorem 5.8 wird außerdem deutlich, dass man diese Variable nicht einsparen kann.

Theorem 5.3 *Das Problem* FPL$[\bot, \mathrel{-\!\!\triangleright}, 1]$-FA *ist* P*-hart.*

Beweis. Wir zeigen $\overline{\text{ASGEP}} \leq_{\text{m}}^{\log}$ FPL$[\bot, \mathrel{-\!\!\triangleright}, 1]$-FA. Dabei bezeichnet $\overline{\text{ASGEP}}$ das Komplement von ASGEP. Da P unter Komplementierung abgeschlossen ist, folgt die P-Härte mit Theorem 2.21.
Es sei $\langle G, s, t\rangle$ eine ASGEP-Instanz und $G = (V, E)$ bestehe aus den m Schichten $V_1, V_2, \ldots, V_m$. Wir konstruieren eine FPL$[\bot, \mathrel{-\!\!\triangleright}, 1]$-FA-Instanz $\langle \alpha_G, \mathcal{M}_G, s\rangle$, so dass

$$\langle G, s, t\rangle \notin \text{ASGEP} \iff \mathcal{M}_G, s \models \alpha_G$$

gilt. Dabei ist p die Variable, die wir in FPL$[\bot, \mathrel{-\!\!\triangleright}, 1]$ verwenden. Der Rahmen $(V, \prec)$ von $\mathcal{M}_G$ ist die pseudotransitive Hülle von G (Definition 2.23). Um t von den anderen Welten in der obersten Schicht V_m zu unterscheiden, benutzen wir p, indem wir die Belegungsfunktion $\xi(p) := \{t\}$ setzen. Damit ist das Modell $\mathcal{M}_G := (V, \prec, \xi, \{p\})$ konstruiert. Ein Beispiel ist in Abbildung 5.2 zu sehen.
Die *aPfad*-Eigenschaft drücken wir mit den Formeln $\alpha_m, \alpha_{m-1}, \ldots, \alpha_1$ aus. Wir definieren sie mit Hilfe der Formeln $\gamma_m, \gamma_{m-1}, \ldots, \gamma_1$, die ein Distanzmaß für die Schichten sind:

$$\begin{array}{llllll} \gamma_m & := & \bot\,, & \alpha_m & := & p\,, \\ \gamma_i & := & \top \mathrel{-\!\!\triangleright} \gamma_{i+1}\,, & \alpha_i & := & \alpha_{i+1} \mathrel{-\!\!\triangleright} \gamma_{i+1} \quad \text{für } i = m-1, m-2, \ldots, 1\,. \end{array}$$

Die Alternierung simulieren wir ähnlich wie im Beweis von Theorem 5.1 und geben jetzt eine zu Behauptung 5.1 ähnliche Behauptung an. Hier benötigen wir die Komplement-Eigenschaften der Formeln (Behauptung 5.1(2)) nicht mehr, da sich die Welten nicht selbst sehen. Eine Welt hat also keine Nachbarn in der eigenen Schicht. Die Funktionsweise dieser Formeln wird in Abbildung 5.2 illustriert.

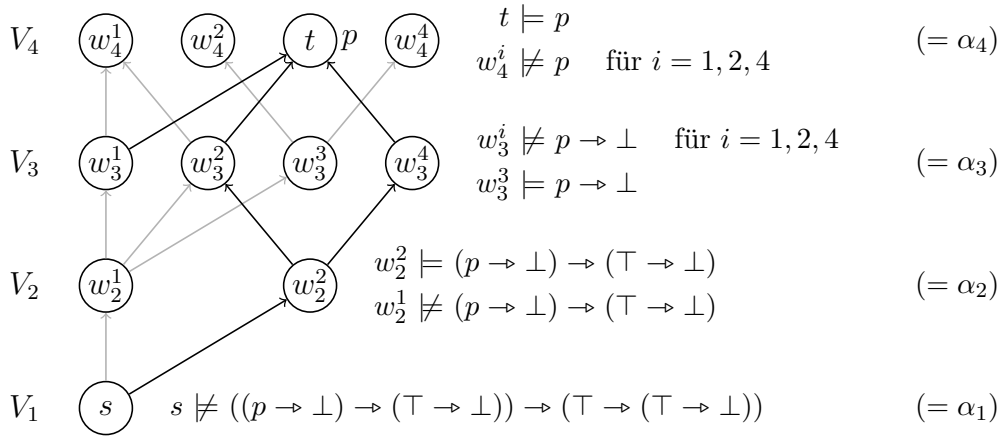

Abbildung 5.2: Ein alternierender Schichtgraph G und dessen Auffassung als Modell $\mathcal{M}_G$. Rechts neben den Schichten stehen die α-Formeln und es wird gezeigt, wie sie die *aPfad*-Eigenschaft darstellen. Die Namen der Welten stehen in den Welten und das p rechts von Welt t ist die Belegung. Die transitive Kanten sind nicht abgebildet. Die schwarzen Kanten sind Bestandteil eines alternierenden Pfades zur Welt t, die grauen nicht. In den Schichten V_1 und V_3 gilt die entsprechende α-Formel genau in den Welten nicht, von denen es einen alternierenden Pfad zur Welt t gibt. In den Schichten V_2 und V_4 ist dies umgekehrt.

Behauptung 5.3 *Für $i = m, m-1, \ldots, 2$ sind folgende Eigenschaften erfüllt:*

(1) $\forall w \in V : \mathcal{M}_G, w \models \gamma_i \iff w \in V_{\geq i+1}$.

(2) $\forall w \in V_{i-1} : \mathcal{M}_G, w \not\models \alpha_{i-1} \iff \exists v \in V_i, w \prec v : \mathcal{M}, v \models \alpha_i$.

Beweis der Behauptung. Wir beweisen den ersten Teil mit Induktion über i. Der Induktionsanfang $i = m$ ist wegen $\gamma_m = \bot$ und $V_{\geq m+1}$ trivial. Für den Induktionsschritt seien $w \in V$ und $m > i \geq 2$, dann gelten folgende Äquivalenzen:

$$\begin{aligned}
&\mathcal{M}_G, w \models \gamma_i \quad (= \top \rightarrow \gamma_{i+1}) && \text{(i)}\\
\Leftrightarrow\ &\forall v \in V, w \prec v : \mathcal{M}_G, v \models \gamma_{i+1} && \text{(ii)}\\
\Leftrightarrow\ &\forall v \in V, w \prec v : v \in V_{\geq i+2} && \text{(iii)}\\
\Leftrightarrow\ &w \in V_{\geq i+1} \ . && \text{(iv)}
\end{aligned}$$

Die Definition von $\rightarrow$ liefert die Äquivalenz zwischen (i) und (ii), die Induktions-

voraussetzung die zwischen (ii) und (iii) und die Konstruktion (und die Irreflexivität) von $\mathcal{M}_G$ die zwischen (iii) und (iv).
Für (2) sei $w \in V_{i-1}$ mit $m \geq i \geq 2$, dann gelten die folgenden Äquivalenzen:

$$\begin{aligned}
&\mathcal{M}_G, w \not\models \alpha_{i-1} \quad (= \alpha_i \twoheadrightarrow \gamma_i) && \text{(i)}\\
\Leftrightarrow\ &\exists v \in V, w \prec v : \mathcal{M}_G, v \models \alpha_i \text{ und } \mathcal{M}_G, v \not\models \gamma_i && \text{(ii)}\\
\Leftrightarrow\ &\exists v \in V_i, w \prec v : \mathcal{M}_G, v \models \alpha_i\ . && \text{(iii)}
\end{aligned}$$

Die Äquivalenz zwischen (i) und (ii) folgt wieder aus der Definition von $\twoheadrightarrow$ und (ii) und (iii) sind wegen (1) äquivalent. ▪
Ähnlich wie in Behauptung 5.2 gibt es auch hier einen Zusammenhang zwischen den α-Formeln und der *aPfad*-Eigenschaft.

Behauptung 5.4 *Für alle $i = m, m-1, \ldots, 1$ und alle Welten $w \in V_i$ gilt:*

$$aPfad_G(w,t) \iff \begin{cases} \mathcal{M}_G, w \models \alpha_i, & \textit{falls} \quad i \textit{ gerade} \\ \mathcal{M}_G, w \not\models \alpha_i, & \textit{falls} \quad i \textit{ ungerade}\ . \end{cases}$$

Beweis der Behauptung. Auch diese Behauptung beweisen wir mittels vollständiger Induktion über i. Die folgenden trivialen Äquivalenzen zeigen den Induktionsanfang für $i = m$ und $w \in V_m$:

$$\begin{aligned}
&aPfad_G(w,t)\\
\Leftrightarrow\ &w = t\\
\Leftrightarrow\ &\mathcal{M}_G, w \models p \quad (= \alpha_m)\ .
\end{aligned}$$

Der Induktionsschritt $i < m$ ist mit Hilfe von Behauptung 5.3 ähnlich zum Induktionsschritt im Beweis von Behauptung 5.2. Wesentlicher Unterschied ist hier, dass die Rollen von geraden und ungerade Schichten vertauscht sind. Wir nehmen erst an, dass i gerade ist, die Schicht V_i besteht also aus $\forall$-Knoten. Für $w \in V_i$ gelten dann folgende Äquivalenzen:

$$\begin{aligned}
&aPfad_G(w,t) && \text{(i)}\\
\Leftrightarrow\ &\forall u, (w,u) \in E : aPfad_G(u,t) && \text{(ii)}\\
\Leftrightarrow\ &\forall u \in V_{i+1}, w \prec u : \mathcal{M}_G, u \not\models \alpha_{i+1} && \text{(iii)}\\
\Leftrightarrow\ &\mathcal{M}_G, w \models \alpha_i\ . && \text{(iv)}
\end{aligned}$$

Aufgrund der Definition der *aPfad*-Eigenschaft (Definition 2.18), sind (i) und (ii) äquivalent. Die Induktionsvoraussetzung zusammen mit der Konstruktion von $\mathcal{M}_G$ liefert die Äquivalenz zwischen (ii) und (iii) und aus Behauptung 5.3(2) folgt die Äquivalenz von (iii) und (iv).
Jetzt sei i ungerade, V_i also eine $\exists$-Schicht. Die folgenden Äquivalenzen gelten dann für $w \in V_i$:

$$
\begin{aligned}
& aPfad_G(w,t) && \text{(i)}\\
\Leftrightarrow\quad & \exists u,(w,u)\in E : aPfad_G(u,t) && \text{(ii)}\\
\Leftrightarrow\quad & \exists u\in V_{i+1}, w\prec u : \mathcal{M}_G,u\models\alpha_{i+1} && \text{(iii)}\\
\Leftrightarrow\quad & \mathcal{M}_G,w\not\models\alpha_i\ . && \text{(iv)}
\end{aligned}
$$

Wie auch schon im Fall, dass i gerade ist, folgt die erste Äquivalenz aus der Definition von *aPfad*, die zweite aus der Induktionsvoraussetzung zusammen mit der Konstruktion von $\mathcal{M}_G$ und die dritte aus Behauptung 5.3(2). ▪
Wir setzen jetzt $\alpha_G := \alpha_1$ und mit Behauptung 5.4 folgt dann

$$\langle G,s,t\rangle\notin \textsc{Asgep} \iff \mathcal{M}_G,s\models\alpha_G\ .$$

Da sich $\mathcal{M}_G$ und α_G aus G in logarithmischem Platz konstruieren lassen,[1] gilt $\overline{\textsc{Asgep}}\leq_{\mathrm{m}}^{\log}$ FPL$[\bot,\twoheadrightarrow,1]$-FA. Die P-Härte von FPL$[\bot,\twoheadrightarrow,1]$-FA folgt mit Theorem 2.21. □

Da jede FPL$[\bot,\twoheadrightarrow,1]$-FA-Instanz auch eine Instanz von BPL$[\bot,\twoheadrightarrow,1]$-FA ist, kann Theorem 5.3 direkt auf BPL übertragen werden.

Bemerkung 5.4 *Das Problem* BPL$[\bot,\twoheadrightarrow,1]$-FA *ist* P*-hart.*

5.1.3 BPL und KC mit beschränkter Variablenzahl

In diesem Abschnitt soll es nun um Fragmente gehen, bei denen nicht nur die intuitionistische Implikation zugelassen ist. Wir wissen aus Bemerkung 5.4, dass das Formelauswerten in BPL bereits P-hart ist, wenn wir nur die Implikation und eine Variable verwenden. Jetzt verzichten wir auch noch auf die letzte Variable und verwenden dafür den Operator $\vee$. Das Prinzip des Beweises der P-Härte von BPL$[\bot,\vee,\twoheadrightarrow,0]$-FA ist ähnlich zu dem der letzten beiden Beweise (Theoreme 5.3 und 5.1). Wieder konstruieren wir aus einem alternierenden Schichtgraphen ein Modell, dass in der letzten Schicht die Zielwelt eines alternierenden Pfades hat. Um diese Zielwelt von anderen Welten in ihrer Schicht unterscheiden zu können, verwenden wir die Eigenschaft von BPL-Modellen, dass Welten sich selbst sehen können, aber nicht müssen. Sieht sich eine Welt selbst, dann erfüllt sie $\top\twoheadrightarrow\bot$ nicht, sieht sie sich nicht (und auch keine anderen Welten), dann ist diese Formel dort erfüllt.

Theorem 5.5 *Das Problem* BPL$[\bot,\vee,\twoheadrightarrow,0]$-FA *ist* P*-hart.*

Beweis. Wie auch schon im Beweis von Theorem 5.3 geben wir eine Reduktion auf das Komplement von Asgep ($\overline{\textsc{Asgep}}\leq_{\mathrm{m}}^{\log}$ BPL$[\bot,\vee,\twoheadrightarrow,0]$-FA) an. Damit folgt aus Theorem 2.21 sofort die P-Härte von BPL$[\bot,\vee,\twoheadrightarrow,0]$-FA. Der Beweis gliedert sich

[1] Die Länge von α_1 ist in etwa die Summe der Längen aller γ-Formeln, also ungefähr m^2.

in zwei Teile. Im ersten Teil verwenden wir die Konstruktion aus dem Beweis von Theorem 5.3 und modifizieren die α-Formeln so, dass sie $\bot$ nicht mehr enthalten. Dafür benutzen wir die beiden Variablen p_1 und p_2, wobei p_2 die Rolle von $\bot$ übernehmen wird. Im zweiten Teil benötigen wir $\bot$-freie Formeln und nutzen eine Technik von Rybakov [Ryb06, Lemma 8], um die beiden Variablen zu ersetzen.
Es sei $\langle G, s, t\rangle$ eine ASGEP-Instanz, wobei $G = (V, E)$ aus m Schichten besteht. Weiter sei $(V, \prec)$ die pseudotransitive Hülle von G gemäß Definition 2.23. Daraus ergibt sich das Modell $\mathcal{M} := (V, \prec, \xi, \{p_1, p_2\})$ mit $\xi(p_1) := \{t\}$ und $\xi(p_2) := \varnothing$. Da p_2 in keiner Welt aus $\mathcal{M}$ erfüllt ist, spielt es in diesem Modell die Rolle von $\bot$. Ähnlich wie im Beweis von Theorem 5.3 definieren wir wieder Formeln, die als Distanzmaß dienen (analog zu den Behauptungen 5.3 und 5.4). Statt $\bot$ verwenden wir jetzt p_2:

$$\begin{array}{lcllcll} \gamma_m & := & p_2 \ , & & \alpha_m & := & p_1 \ , \\ \gamma_i & := & \top \twoheadrightarrow \gamma_{i+1} \ , & & \alpha_i & := & \alpha_{i+1} \twoheadrightarrow \gamma_{i+1} \quad \text{für } i = m-1, m-2, \ldots, 1 \ . \end{array}$$

Der Zusammenhang zwischen einem alternierenden Pfad in G und α_1 gilt hier wie auch schon im Beweis von Theorem 5.3 auf Basis der Behauptungen 5.3 und 5.4.

Behauptung 5.5 *Es gilt* $\mathcal{M}, s \models \alpha_1 \iff \langle G, s, t\rangle \notin$ ASGEP .

Im Folgenden wollen wir jedes Vorkommen der Variablen p_1 und p_2 durch Formeln ohne Variablen ersetzen. Dabei nutzen wir eine Konstruktion, die bereits Rybakov [Ryb06, Lemma 8] verwendet hat. Da in den Modellen im weiteren Verlauf dieses Beweises keine Variablen mehr vorkommen (wir betrachten BPL$[\bot, \vee, \twoheadrightarrow, 0]$ ab hier), geben wir Modelle nur noch als Paar bestehend aus einer Weltenmenge und der zugehörigen Sichtbarkeitsrelation an. Wir definieren zunächst drei Modelle $\mathcal{M}_k = (W_k, R_k)$ für $k = 1, 2, 3$ wie folgt:

$$\begin{array}{lcl} W_k & := & \{b_k, a_1^k, a_2^k, \ldots, a_{k+2}^k\} \ , \\ R_k & := & \{(b_k, b_k), (a_{k+2}, b_k)\} \cup \{(a_i^k, a_j^k) \mid k+2 \geq i > j \geq 1\} \ . \end{array}$$

Diese drei Modelle sind als Teilmodelle in Abbildung 5.3 zu sehen. In der weiteren Konstruktion werden sie verwendet, um die ausgezeichnete Zielwelt eines alternierenden Schichtgraphen von den anderen Welten in der obersten Schicht unterscheidbar zu machen. Wir geben jetzt ein Modell $\mathcal{M}_G = (W, S)$ an, dass sich aus $\mathcal{M}_1$, $\mathcal{M}_2$, $\mathcal{M}_3$ und $\mathcal{M}$ wie folgt zusammensetzt:

$$\begin{array}{lcl} W & := & W_1 \cup W_2 \cup W_3 \cup V \ , \\ S^\xi & := & \{(w, a_3^1), (v, a_4^2), (v, a_5^3) \mid w \in V \setminus \{t\}, v \in V\} \ , \\ S \text{ ist die transitive Hülle von } & & \prec \cup R_1 \cup R_2 \cup R_3 \cup S^\xi \quad . \end{array}$$

Von $\mathcal{M}$ verwenden wir hier nur noch den Rahmen. Ein Beispiel ist in Abbildung 5.3 zu sehen. Da $|W_1 \cup W_2 \cup W_3| = 15$ gilt und $\prec$ bereits transitiv ist, lässt sich

die transitive Hülle für S auch in logarithmischem Platz berechnen. Mit Hilfe der Modelle $\mathcal{M}_1$, $\mathcal{M}_2$ und $\mathcal{M}_3$ und S^ξ simulieren wir die Belegungsfunktion ξ von $\mathcal{M}$. Um jetzt auch die Variablen p_1 und p_2 aus α_1 entfernen zu können, geben wir zwei Formeln φ_1 und φ_2 an, die diese später ersetzen. Dafür definieren wir noch folgende Abkürzungen:

$$\psi_1 := \top \twoheadrightarrow \bot\,, \qquad \psi_3 := \top \twoheadrightarrow (\top \twoheadrightarrow (\top \twoheadrightarrow \bot))\,,$$
$$\psi_2 := \top \twoheadrightarrow (\top \twoheadrightarrow \bot)\,, \qquad \psi_4 := \top \twoheadrightarrow (\top \twoheadrightarrow (\top \twoheadrightarrow (\top \twoheadrightarrow \bot)))\,.$$

Diese Formeln sind die Bestandteile von φ_1 und φ_2:

$$\varphi_1 := (\psi_3 \twoheadrightarrow \psi_2) \twoheadrightarrow ((\psi_2 \twoheadrightarrow \psi_1) \vee \psi_3) \quad \text{und}$$
$$\varphi_2 := (\psi_4 \twoheadrightarrow \psi_3) \twoheadrightarrow ((\psi_3 \twoheadrightarrow \psi_2) \vee \psi_4)\,.$$

Der Zusammenhang zwischen diesen Formeln (bzw. deren Teilformeln) und dem zusammengesetzten Modell $\mathcal{M}_G$ ist in Abbildung 5.3 dargestellt.

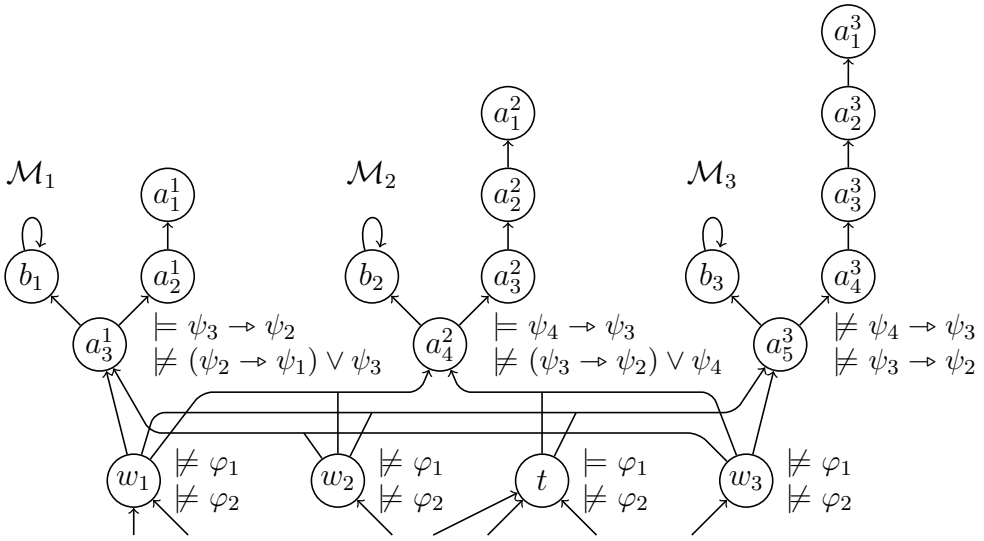

Abbildung 5.3: Hier sind die 3 Modelle $\mathcal{M}_1$, $\mathcal{M}_2$ und $\mathcal{M}_3$ dargestellt und wie sie an die oberste Schicht von $\mathcal{M}$ angehängt werden. Transitive Kanten sind nicht abgebildet. Außerdem steht an den Basiswelten von $\mathcal{M}_1$, $\mathcal{M}_2$ und $\mathcal{M}_1$, welche Bestandteile von φ_1 und φ_2 dort erfüllt sind und welche nicht. Damit lässt sich erkennen, in welcher Welt aus $\mathcal{M}$ welche der beiden Formeln φ_1 und φ_2 erfüllt (bzw. nicht erfüllt) ist.

Die wesentliche Eigenschaft der Konstruktion lässt sich wie folgt charakterisieren: Ist von einer Welt w aus V die Basiswelt a_3^1 von $\mathcal{M}_1$ sichtbar, dann und nur dann ist φ_1 in w nicht erfüllt. Ist die Basiswelt a_4^2 von $\mathcal{M}_2$ sichtbar, wird φ_2 in w nicht erfüllt. Aus der obersten Schicht von $\mathcal{M}$ sehen alle Welten die Basiswelt a_4^2 von

$\mathcal{M}_2$. Alle Welten außer t sehen zusätzlich a_3^1. Aus technischen Gründen sehen außerdem alle Welten die Basiswelt a_5^3 von $\mathcal{M}_3$. Die folgende Behauptung zeigt den Zusammenhang zwischen $\mathcal{M}$ und $\mathcal{M}_G$.

Behauptung 5.6 *Es sei δ eine* BPL$[\vee, \rightarrow\!\!\!\!\cdot, 2]$*-Formel mit den beiden Variablen p_1 und p_2. Die Formel δ_G entsteht aus δ, indem jedes Vorkommen von p_1 (bzw. p_2) durch φ_1 (bzw. φ_2) ersetzt wird ($\delta_G := \delta[p_1/\varphi_1][p_2/\varphi_2]$). Dann gilt für alle Welten $w \in V$*

$$\mathcal{M}, w \models \delta \iff \mathcal{M}_G, w \models \delta_G \ .$$

Beweis der Behauptung. Wir beweisen diese Behauptung mit vollständiger Induktion über den Aufbau von δ. Da wir die beiden Variablen p_1 und p_2 und die beiden Formeln φ_1 und φ_2 verwenden, ist in diesem Beweis der Index i immer 1 oder 2.
Für den Induktionsanfang sei $\delta = p_i$, also ist $\delta_G = \varphi_i$.
Falls $\mathcal{M}, w \not\models p_i$ gilt, sieht w nach Konstruktion die Welt a_{i+2}^i im Modell $\mathcal{M}_G$. Nur in der Welt t aus $\mathcal{M}$ ist die Variable p_1 erfüllt und diese Welt sieht als Einzige die Welt a_3^1 nicht. Aus der Konstruktion folgt dann $\mathcal{M}_G, w \not\models \varphi_i$.
Für die andere Richtung gelte $\mathcal{M}_G, w \not\models \varphi_i$. Es muss also eine Welt $v \in W$ mit $(w, v) \in S$ und

$$\mathcal{M}, v \models \psi_{i+2} \rightarrow\!\!\!\!\cdot\ \psi_{i+1} \quad \text{und} \quad \mathcal{M}, v \not\models (\psi_{i+1} \rightarrow\!\!\!\!\cdot\ \psi_i) \vee \psi_{i+2}$$

geben. Wir nehmen jetzt an, dass diese Welt v aus V ist (also auch in dem Modell $\mathcal{M}$ vorkommt). Für alle Welten $u \in V$ gilt nach Konstruktion $(u, a_5^3) \in S$ und es gilt weiter $\mathcal{M}_G, a_5^3 \not\models \psi_4 \rightarrow\!\!\!\!\cdot\ \psi_3$ und $\mathcal{M}_G, a_5^3 \not\models \psi_3 \rightarrow\!\!\!\!\cdot\ \psi_2$. Somit ist die Annahme falsch und v kann nur aus $W_1 \cup W_2 \cup W_3$ sein. Aus der Konstruktion der Modelle $\mathcal{M}_1$, $\mathcal{M}_2$ und $\mathcal{M}_3$ folgt, dass

$$\begin{aligned} &\mathcal{M}_j, v \models \psi_{i+2} \rightarrow\!\!\!\!\cdot\ \psi_{i+1} \text{ und } \mathcal{M}_j, v \not\models (\psi_{i+1} \rightarrow\!\!\!\!\cdot\ \psi_i) \vee \psi_{i+2} \\ &\Leftrightarrow \quad i = j \text{ und } v = a_{i+2}^i \end{aligned}$$

gilt. Daraus folgt, dass v nur a_{i+2}^i sein kann. Nach der Konstruktion gilt $(w, a_{i+2}^i) \in S$ genau dann, wenn $w \notin \xi(p_i)$, also folgt $\mathcal{M}, w \not\models p_i$.
Der Induktionsschritt für $\delta = \delta^1 \vee \delta^2$ ist offensichtlich klar. Es sei deshalb jetzt $\delta = \delta^1 \rightarrow\!\!\!\!\cdot\ \delta^2$ und damit ist $\delta_G = \delta_G^1 \rightarrow\!\!\!\!\cdot\ \delta_G^2$.
Im Falle $\mathcal{M}, w \not\models \delta$ gibt es eine Welt $v \in V$ mit $w \prec v$ und $\mathcal{M}, v \models \delta^1$ und $\mathcal{M}, v \not\models \delta^2$. Nach Induktionsvoraussetzung gilt $\mathcal{M}_G, v \models \delta_G^1$ und $\mathcal{M}_G, v \not\models \delta_G^2$ und es folgt $\mathcal{M}_G, w \not\models \delta_G$.
Für die andere Richtung gelte nun $\mathcal{M}_G, w \not\models \delta_G$. Es muss also eine Welt $v \in W$ mit $\mathcal{M}_G, v \models \delta_G^1$ und $\mathcal{M}_G, v \not\models \delta_G^2$ und $(w, v) \in S$ geben. Man kann sich leicht davon überzeugen, dass für alle Welten $u \in W_1 \cup W_2 \cup W_3$ immer $\mathcal{M}_G, u \models \varphi_i$ gilt. Da in δ_G^2 keine Negationen außerhalb der φ_i-Teilformeln vorkommen und alle Vorkommen von φ_i in u erfüllt sind, ist auch δ_G^2 in u erfüllt. Die Welt v muss somit in $W \setminus (W_1 \cup W_2 \cup W_3) = V$ liegen. Nach Induktionsvoraussetzung gilt $\mathcal{M}, v \models \delta^1$ und $\mathcal{M}, v \not\models \delta^2$ und nach Konstruktion $w \prec v$. Also folgt $\mathcal{M}, w \not\models \delta$. ∎

Auf ähnliche Weise wird Lemma 8 in [Ryb06] bewiesen. Wir setzen nun

$$\alpha_G \quad := \quad \alpha_1[p_1/\varphi_1][p_2/\varphi_2]$$

und aus den Behauptungen 5.5 und 5.6 folgt

$$\langle G, s, t\rangle \notin \textsc{Asgep} \quad \Longleftrightarrow \quad \mathcal{M}_G, s \models \alpha_G \ .$$

Die Konstruktion von $\mathcal{M}_G$ und α_G aus G ist in logarithmischem Platz möglich und es gilt daher $\overline{\textsc{Asgep}} \leq_{\mathrm{m}}^{\log} \mathrm{FPL}[\bot, \rightarrow\!\!\!\!\triangleright, 1]\text{-FA}$. Mit Theorem 2.21 folgt dann, dass $\mathrm{FPL}[\bot, \rightarrow\!\!\!\!\triangleright, 1]\text{-FA}$ P-hart ist. □

Beschränkt man für die Logiken LC, KC und IPL die Zahl der Variablen, haben wir bereits eine Reihe Ergebnisse für die Komplexität des Formelauswertungsproblems angegeben. Aus den Theoremen 3.3, 3.10 und 3.11 folgt, dass es für LC mit beschränkter Variablenzahl NC^1-vollständig ist. Für KC und IPL ohne Variablen folgt dies aus Theorem 2.31. Lässt man eine Variable bei KC zu, so liefert Theorem 4.30 die NC^1-Vollständigkeit und nach Theorem 4.29 ist die Formelauswertung für IPL[1] AC^1-vollständig. Offen ist nun noch die Frage, was passiert, wenn man bei KC und IPL mehr als eine Variable zulässt.

Theorem 5.6 *Das Problem* $\mathrm{KC}[\wedge, \vee, \rightarrow\!\!\!\!\triangleright, 2]\text{-FA}$ *ist* P*-hart.*

Beweis. Wir zeigen $\mathrm{IPL}[\rightarrow\!\!\!\!\triangleright]\text{-FA} \leq_{\mathrm{m}}^{\log} \mathrm{KC}[\wedge, \vee, \rightarrow\!\!\!\!\triangleright, 2]\text{-FA}$. Mit Bemerkung 5.2 folgt die P-Härte für $\mathrm{KC}[\wedge, \vee, \rightarrow\!\!\!\!\triangleright, 2]\text{-FA}$. Die wesentlichen Details der Konstruktion aus diesem Beweis verwendete bereits Rybakov [Ryb06, Theorem 4], um die PSPACE-Vollständigkeit von $\mathrm{IPL}[\wedge, \vee, \rightarrow\!\!\!\!\triangleright, 2]$-TAUT zu zeigen.
In einem ersten Schritt konstruieren wir Formeln mit zwei Variablen, mit denen wir später die Variablen in Formeln mit beliebig vielen Variablen ersetzen. Wir nennen diese Formeln *Ersetzungsformeln*. Als nächstes geben wir Modelle an, die für jede Ersetzungsformel eine eindeutig bestimmte maximale Welt haben, in der die Formel nicht erfüllt wird. In allen echten Nachfolgern dieser Welt wird die Formel erfüllt. Diese Modelle bezeichnen wir als *generische Modelle*. Zuletzt geben wir die Transformation einer $\mathrm{IPL}[\rightarrow\!\!\!\!\triangleright]$-FA-Instanz in eine $\mathrm{KC}[\wedge, \vee, \rightarrow\!\!\!\!\triangleright, 2]$-FA-Instanz an. Dabei ersetzen wir die Variablen der $\mathrm{IPL}[\rightarrow\!\!\!\!\triangleright]$-Formel durch die Ersetzungsformeln. Als $\mathrm{KC}[\wedge, \vee, \rightarrow\!\!\!\!\triangleright, 2]$-Modell verwenden wir eine Vereinigung von einem bestimmten generischen Modell und dem gegebenen $\mathrm{IPL}[\rightarrow\!\!\!\!\triangleright]$-Modell. Wir verändern die Belegungsfunktion des $\mathrm{IPL}[\rightarrow\!\!\!\!\triangleright]$-Modells so, dass das aus der Vereinigung resultierende Modell ein $\mathrm{KC}[\wedge, \vee, \rightarrow\!\!\!\!\triangleright, 2]$-Modell ist.
Insgesamt ist die Konstruktion – insbesondere der Anfang der induktiven Definition der Ersetzungsformeln – sehr technisch. Als Variablen in $\mathrm{KC}[\wedge, \vee, \rightarrow\!\!\!\!\triangleright, 2]$ verwenden wir p_1 und p_2. Abbildung 5.4 zeigt den Anfang eines jeden generischen Modells, die weitere Konstruktion kann man aus Abbildung 5.5 ableiten. Wir geben an, welches die maximalen Welten sind, die die Ersetzungsformeln nicht erfüllen. Die Existenz dieser Welten ist wesentlich für die Transformation einer $\mathrm{IPL}[\rightarrow\!\!\!\!\triangleright]$-FA-Instanz in eine $\mathrm{KC}[\wedge, \vee, \rightarrow\!\!\!\!\triangleright, 2]$-FA-Instanz.

Konstruktion der Ersetzungsformeln. Die folgenden Formeln bilden die Basis der induktiven Definition der Ersetzungsformeln:

$$\begin{array}{lcllcllcl}
\delta_1 & := & p_1 \rightarrowtriangle p_2 \ , & \delta_2 & := & p_2 \rightarrowtriangle p_1 \ , & \delta_3 & := & p_1 \vee p_2 \ , \\
\varepsilon_1 & := & \delta_2 \rightarrowtriangle (\delta_1 \vee \delta_3) \ , & \varepsilon_3 & := & \delta_1 \rightarrowtriangle (\delta_2 \vee \delta_3) \ , & & & \\
\varepsilon_2 & := & \delta_3 \rightarrowtriangle (\delta_1 \vee \delta_2) \ , & \varepsilon_4 & := & (\varepsilon_1 \wedge \varepsilon_2 \wedge \varepsilon_3) \rightarrowtriangle (\delta_1 \vee \delta_2 \vee \delta_3) \ . & & &
\end{array}$$

Mit Hilfe dieser Formeln geben wir die ersten Ersetzungsformeln an:

$$\begin{array}{lcllcl}
\alpha_1^1 & := & (\varepsilon_1 \wedge \varepsilon_2) \rightarrowtriangle (\varepsilon_3 \vee \varepsilon_4) \ , & \beta_1^1 & := & (\varepsilon_2 \wedge \varepsilon_3) \rightarrowtriangle (\varepsilon_1 \vee \varepsilon_4) \ , \\
\alpha_2^1 & := & (\varepsilon_1 \wedge \varepsilon_3) \rightarrowtriangle (\varepsilon_2 \vee \varepsilon_4) \ , & \beta_2^1 & := & (\varepsilon_2 \wedge \varepsilon_4) \rightarrowtriangle (\varepsilon_1 \vee \varepsilon_3) \ , \\
\alpha_3^1 & := & (\varepsilon_1 \wedge \varepsilon_4) \rightarrowtriangle (\varepsilon_2 \vee \varepsilon_3) \ , & \beta_3^1 & := & (\varepsilon_3 \wedge \varepsilon_4) \rightarrowtriangle (\varepsilon_1 \vee \varepsilon_2) \ .
\end{array}$$

Wir bezeichnen den oberen Index als *Level* und definieren die Ersetzungsformeln der nächsten Level induktiv. Für jedes Level k benötigen wir zunächst noch ein technisches Hilfsmittel. Dafür seien $n_1 := 3$ und $n_{k+1} := |P_k|$, wobei $P_k := \{(x, y) \mid 2 \leq x, y \leq n_k\}$ ist. Mit Induktion über k kann man leicht $n_{k+1} = (n_k - 1)^2$ zeigen. In Level k definieren wir die Ersetzungsformeln α_i^k und β_i^k für $1 \leq i \leq n_k$. Für den Übergang von Level k zu Level $k+1$ brauchen wir eine Funktion $\langle \cdot, \cdot \rangle_k : P_k \mapsto \{1, 2, \ldots, (n_k - 1)^2\}$, die sich leicht berechnen lässt. Auch die Umkehrung dieser Funktion muss leicht berechenbar sein. Hier kann man zum Beispiel $\langle i, j \rangle_k := (j - 1) + (n_k - 1) \cdot (i - 2)$ für $2 \leq i, j \leq n_k$ verwenden. Damit können wir die Ersetzungsformeln aus Level $k + 1$ für $k \geq 1$ angeben. Für $i, j \in \{2, 3, \ldots, n_k\}$ definieren wir

$$\begin{array}{lcl}
\alpha_{\langle i,j \rangle_k}^{k+1} & := & \alpha_1^k \rightarrowtriangle (\beta_1^k \vee \alpha_i^k \vee \beta_j^k) \ , \\
\beta_{\langle i,j \rangle_k}^{k+1} & := & \beta_1^k \rightarrowtriangle (\alpha_1^k \vee \alpha_i^k \vee \beta_j^k) \ .
\end{array}$$

Konstruktion der generischen Modelle. Für $t \geq 1$ definieren wir die generischen Modelle $\mathcal{M}_t^{gen} := (W_t^{gen}, S_t^{gen}, \xi^{gen}, \{p_1, p_2\})$ wie folgt. Zuerst geben wir die Menge der Welten an und dann konstruieren wir die Sichtbarkeitsrelation. Den Abschluss bildet die Belegungsfunktion. Für die Welten definieren wir

$$\begin{array}{lcll}
W_0 & := & \{c, d_1, d_2, d_3, e_1, e_2, e_3, e_4\} \ , & \\
W_k & := & \{a_i^k, b_i^k \mid 1 \leq i \leq n_k\} & \text{für } 1 \leq k \leq t \qquad (n_k \text{ ist wie oben definiert})
\end{array}$$

und damit

$$W_t^{gen} := \bigcup_{l=0}^{t} W_l \ .$$

Der Index k bezeichnet das Level, zu dem eine Welt gehört. Die Sichtbarkeitsrelation geben wir in verschiedenen Abschnitten an. Abbildung 5.4 zeigt die Welten aus W_0 und W_1 und den zu diesen Welten gehörende Teil S^{top} von S_t^{gen}. Formal ist S^{top} die transitive und reflexive Hülle von

$$
\begin{aligned}
& \{(d_1, c), (d_2, c), (d_3, c)\} \\
\cup \quad & \{(e_1, d_1), (e_1, d_3)\} \cup \{(e_2, d_1), (e_2, d_2)\} \cup \{(e_3, d_2), (e_3, d_3)\} \\
& \cup \{(e_4, d_1), (e_4, d_2), (e_4, d_3)\} \\
\cup \quad & \{(b_3^1, e_1), (b_3^1, e_2)\} \cup \{(b_2^1, e_1), (b_2^1, e_3)\} \cup \{(b_1^1, e_1), (b_1^1, e_4)\} \\
& \cup \{(a_3^1, e_2), (a_3^1, e_3)\} \cup \{(a_2^1, e_2), (a_2^1, e_4)\} \cup \{(a_1^1, e_3), (a_1^1, e_4)\} \ .
\end{aligned}
$$

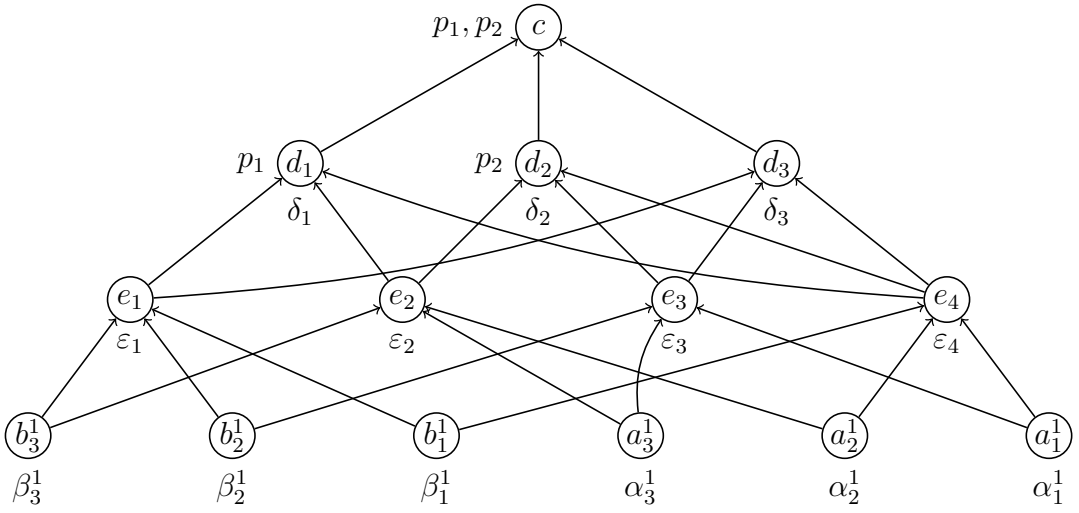

Abbildung 5.4: Dies ist die Spitze aller generischen Modelle bestehend aus den Welten aus W_0 und W_1 und der Sichtbarkeitsrelation S^{top}. Steht unter einer Welt w eine Formel φ, bedeutet dies, dass w die größte Welt ist, in der φ nicht erfüllt ist. Die Belegungsfunktion wird durch die links der Welten stehenden Variablen p_1 und p_2 dargestellt. Reflexive und transitive Kanten sind nicht abgebildet.

Für die Welten aus den Leveln ≥ 2 wird die Sichtbarkeitsrelation im Folgenden definiert. Wir setzen für $k \geq 1$

$$
\begin{aligned}
S^a_{k+1} &:= \{(a^{k+1}_{\langle i,j\rangle_k}, b^k_1), (a^{k+1}_{\langle i,j\rangle_k}, a^k_i), (a^{k+1}_{\langle i,j\rangle_k}, b^k_j) \mid 2 \leq i, j \leq n_k\} \ , \\
S^b_{k+1} &:= \{(b^{k+1}_{\langle i,j\rangle_k}, a^k_1), (b^{k+1}_{\langle i,j\rangle_k}, a^k_i), (b^{k+1}_{\langle i,j\rangle_k}, b^k_j) \mid 2 \leq i, j \leq n_k\} \ .
\end{aligned}
$$

Für $t = 1$ ist $S^{gen}_t := S^{top}$, für $t > 1$ definieren wir

$$
S' := S^{top} \cup \bigcup_{l=2}^{t} (S^a_l \cup S^b_l) \ .
$$

Da die Sichtbarkeitsrelation transitiv sein muss, sich aber die transitive Hülle von

S' nicht in logarithmischem Platz berechnen lässt, benutzen wir wieder die Idee der pseudotransitiven Hülle (Definition 2.23). Dabei wird jede Welt aus Level k für $k \geq 2$ mit allen Welten verbunden, die mindestens zwei Level über ihr liegen:

$$T_k \quad := \quad W_k \times (\bigcup_{l=0}^{k-2} W_l) \ .$$

Für $t > 1$ ist T die Vereinigung aller pseudotransitiven Kanten:

$$T \quad := \quad \bigcup_{l=2}^{t} T_l \ .$$

Damit können wir jetzt die Sichtbarkeitsrelation S_t^{gen} von $\mathcal{M}_t^{gen}$ auch für $t > 1$ wie folgt angeben:

S_t^{gen} ist die reflexive Hülle von $T \cup S'$.

Ein Ausschnitt aus $\mathcal{M}_t^{gen}$ ist in Abbildung 5.5 dargestellt.

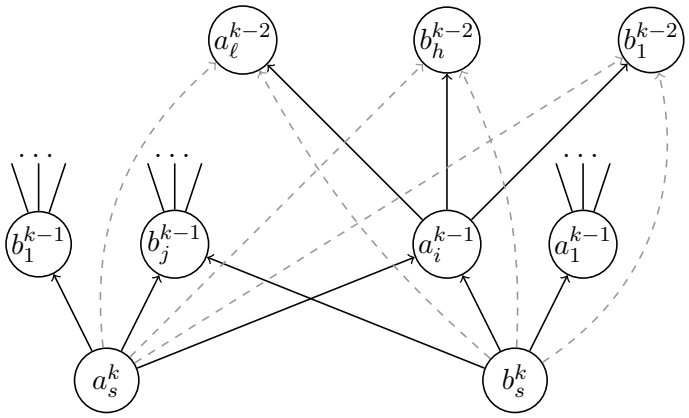

Abbildung 5.5: Dies ist ein Ausschnitt der Level k, $k-1$ und $k-2$ des generischen Modells $\mathcal{M}_t^{gen}$, wobei $s = \langle i, j \rangle_{k-1}$, $i = \langle \ell, h \rangle_{k-2}$ und $k \leq t$ sind. Die grau gestrichelten Kanten sind die pseudotransitiven Kanten zwischen Level k und $k-2$. Wie wir in Behauptung 5.7 zeigen, ist zum Beispiel a_i^{k-1} die größte Welt, in der α_i^{k-1} nicht erfüllt ist. Die reflexiven Kanten sind nicht dargestellt.

Die Belegungsfunktion ξ^{gen} (siehe auch Abbildung 5.4) ist für alle generischen Modell $\mathcal{M}_t^{gen}$ wie folgt definiert:

$$\xi^{gen}(p) \quad := \quad \{c, d_1\} \ ,$$
$$\xi^{gen}(q) \quad := \quad \{c, d_2\} \ .$$

Wie bereits erwähnt, ist das Ziel dieser Konstruktion, dass es für jede Ersetzungsformel genau eine maximale Welt gibt, in der sie nicht erfüllt wird. In allen echten Nachfolgern dieser maximalen Welt und in allen Welten, die sie nicht sehen, wird die entsprechende Ersetzungsformel erfüllt. Die Bezeichnungen sind so gewählt, dass die Welt a_i^k (bzw. b_i^k) diese maximale Welt für die Formel α_i^k (bzw. β_i^k) ist. Das bedeutet α_i^k (bzw. β_i^k) wird von keiner Welt erfüllt, die a_i^k (bzw. b_i^k) sieht.

Behauptung 5.7 *Es seien $t \geq 1$ und w eine Welt aus $\mathcal{M}_t^{gen}$. Dann gilt für alle $k \leq t$ und alle $i \leq n_k$*

$$\mathcal{M}_t^{gen}, w \not\models \alpha_i^k \iff (w, a_i^k) \in S_t^{gen} ,$$
$$\mathcal{M}_t^{gen}, w \not\models \beta_i^k \iff (w, b_i^k) \in S_t^{gen} .$$

Beweis der Behauptung. Wir zeigen diese Behauptung mit vollständiger Induktion über das Level k (vergleiche dazu Lemma 5 in [Ryb06]).
Für den Induktionsanfang sei $k = 1$. Wir schauen uns hier die Formel α_1^1 an, der Beweis für die anderen Formeln aus Level 1 ist analog. Es gilt $\mathcal{M}_t^{gen}, a_1^1 \not\models \alpha_1^1$ wegen $(a_1^1, e_1), (a_1^1, e_2) \notin S_t^{gen}$ und $(a_1^1, e_3), (a_1^1, e_4) \in S_t^{gen}$ (siehe dazu Abbildung 5.4). Also folgt für jede Welt w mit $(w, a_1^1) \in S_t^{gen}$ direkt $\mathcal{M}_t^{gen}, w \not\models \alpha_1^1$. Umgekehrt muss eine Welt w mit $\mathcal{M}_t^{gen}, w \not\models \alpha_1^1$ auch e_3 und e_4 als Nachfolger haben. Da a_1^1 der einzige direkte Vorgänger von e_3 und e_4 ist, gilt auch $(w, a_1^1) \in S_t^{gen}$.
Für den Induktionsschritt gehen wir zu Level k mit $k > 1$. Wir betrachten die Formel α_i^k mit $1 \leq i \leq n_k$, für die β-Formeln dieses Levels verläuft der Beweis analog. Es seien j und ℓ so gewählt, dass $i = \langle j, \ell \rangle_{k-1}$ gilt, also ist $\alpha_i^k = \alpha^k_{\langle j,\ell\rangle_{k-1}} = \alpha_1^{k-1} \twoheadrightarrow (\beta_1^{k-1} \vee \alpha_j^{k-1} \vee \beta_\ell^{k-1})$. Nach Konstruktion gilt $(a^k_{\langle j,\ell\rangle_{k-1}}, x) \in S_t^{gen}$ für $x \in \{b_1^{k-1}, a_j^{k-1}, b_\ell^{k-1}\}$. Also folgt für alle w mit $(w, a_i^k) \in S_t^{gen}$, dass α_i^k in w nicht erfüllt ist. Die Umkehrung gilt wieder, ähnlich wie im Induktionsanfang, da a_i^k der einzige direkte Vorgänger von b_1^{k-1}, a_j^{k-1} und b_ℓ^{k-1} ist. ▪

Reduktion von IPL[$\twoheadrightarrow$]-FA **auf** KC[$\wedge, \vee, \twoheadrightarrow, 2$]-FA**.** Es sei $\langle \varphi, \mathcal{M}, w \rangle$ eine Instanz von IPL[$\twoheadrightarrow$]-FA. Wir konstruieren jetzt aus φ mit Hilfe der Ersetzungsformeln eine KC[$\wedge, \vee, \twoheadrightarrow, 2$]-FA-Formel φ^2 und mit den generischen Modellen aus $\mathcal{M}$ ein KC[$\wedge, \vee, \twoheadrightarrow, 2$]-FA-Modell $\mathcal{M}^2$, so dass

$$\mathcal{M}^2, w \models \varphi_2 \iff \mathcal{M}, w \models \varphi$$

gilt. Im weiteren Verlauf gehen wir davon aus, dass φ die Variablen $q_1, q_2, \ldots, q_m$ enthält und $\mathcal{M} = (W, S, \xi, \mathrm{VAR})$ ist. Wir wählen k so, dass es der kleinste Wert mit $n_k > m$ ist. So ist sicher gestellt, dass es für jede Variable auch eine eigene Ersetzungsformel gibt. Jetzt konstruieren wir φ^2 aus φ, indem wir jedes Vorkommen von jeder Variablen q_i durch $\alpha_i^k \vee \beta_i^k$ ersetzen:

$$\varphi^2 := \varphi[q_1/\alpha_1^k \vee \beta_1^k][q_2/\alpha_2^k \vee \beta_2^k] \ldots [q_m/\alpha_m^k \vee \beta_m^k] .$$

Da $k \leq 1 + \log(m)$ gilt, lässt sich φ^2 aus φ in logarithmischem Platz konstruieren.

Das Modell $\mathcal{M}^2 = (W^2, S^2, \xi^2, \{p_1, p_2\})$ ist eine Vereinigung aus $\mathcal{M}$ und dem generischen Modell $\mathcal{M}_k^{gen}$. Für die Menge W^2 der Welten gilt

$$W^2 \quad := \quad W_k^{gen} \cup W \ .$$

Die Sichtbarkeitsrelation S^2 wird so konstruiert, dass eine Welt w aus W die Welten a_i^k und b_i^k sieht, wenn q_i in w nicht erfüllt ist. Basierend auf Behauptung 5.7 ist damit sichergestellt, dass $\alpha_i^k \vee \beta_i^k$ – die Übersetzung von q_i – in w nicht erfüllt ist. Aus technischen Gründen gibt es auch noch eine Verbindung zwischen jeder Welt aus W und a_{m+1}^k und b_{m+1}^k:

$$S_\xi \quad := \quad \{(w, a_i^k), (w, b_i^k) \mid w \in W \setminus \xi(q_i)\} \cup \{(w, a_{m+1}^k), (w, b_{m+1}^k) \mid w \in W\} \ .$$

Als nächstes stellen wir die Transitivität sicher. Da sich die transitive Hülle wieder nicht in logarithmischem Platz berechnen lässt, greifen wir erneut auf die pseudotransitive Hülle zurück (Definition 2.23). Wir verbinden jede Welt aus W mit jeder Welt aus den Leveln $1, 2, \ldots, k-1$:

$$S^{trans} \quad := \quad W \times \bigcup_{l=0}^{k-1} W_l \ .$$

Jetzt können wir S^2 vollständig angeben:

S^2 ist die reflexive Hülle von $S_k^{gen} \cup S \cup S_\xi \cup S^{trans}$.

Als Belegungsfunktion nutzen wir die der generischen Modelle:

$$\xi^2 \quad := \quad \xi^S \ .$$

Die Belegungsfunktion aus $\mathcal{M}$ wird durch die Verbindungskanten zwischen $\mathcal{M}$ und den Welten in Level k aus $\mathcal{M}_k^{gen}$ simuliert. Da jede Welt aus W^2 die Welt c sieht, ist sichergestellt, dass S^2 eine gerichtete Halbordnung und somit $\mathcal{M}^2$ ein KC$[\wedge, \vee, \rightarrow, 2]$-FA-Modell ist. Im Folgenden zeigen wir, dass diese Konstruktion eine korrekte Reduktion ist in unserem Sinne ist.

Behauptung 5.8 *Für alle Welten $w \in W$ gilt $\mathcal{M}, w \models \varphi \iff \mathcal{M}^2, w \models \varphi^2$.*

Beweis der Behauptung. Wir führen diesen Beweis mit Induktion über den Aufbau von φ (vergleiche dazu Lemma 7 in [Ryb06]).
Für den Induktionsanfang sei $\varphi = q_i$ für $i \in \{1, 2, \ldots, m\}$, demnach ist $\varphi^2 = \alpha_i^k \vee \beta_i^k$. Aus $\mathcal{M}, w \not\models q_i$ folgt in $\mathcal{M}^2$, dass w via S^2 (bzw. S_ξ) mit a_i^k und b_i^k verbunden ist. Also gilt $\mathcal{M}^2, w \not\models \alpha_i^k \vee \beta_i^k$ nach Behauptung 5.7.
Wir gehen jetzt für ein $w \in W$ von $\mathcal{M}^2, w \not\models \alpha_i^k \vee \beta_i^k$ aus. Nach Konstruktion der Ersetzungsformeln gilt

$$\begin{aligned} \alpha_i^k &= \alpha_1^{k-1} \rightarrow (\beta_1^{k-1} \vee \alpha_j^{k-1} \vee \beta_\ell^{k-1}) \quad \text{und} \\ \beta_i^k &= \beta_1^{k-1} \rightarrow (\alpha_1^{k-1} \vee \alpha_j^{k-1} \vee \beta_\ell^{k-1}) \end{aligned}$$

für $i = \langle j, \ell \rangle_{k-1}$. Es muss also Welten $w', w'' \in W^2$ mit $(w, w') \in S^2$ und $(w, w'') \in S^2$ geben, so dass

$$\mathcal{M}^2, w' \models \alpha_1^{k-1} \quad \text{und} \quad \mathcal{M}^2, w' \not\models \beta_1^{k-1} \vee \alpha_j^{k-1} \vee \beta_\ell^{k-1} \quad \text{und}$$
$$\mathcal{M}^2, w'' \models \beta_1^{k-1} \quad \text{und} \quad \mathcal{M}^2, w'' \not\models \alpha_1^{k-1} \vee \alpha_j^{k-1} \vee \beta_\ell^{k-1}$$

gilt. Modell $\mathcal{M}^2$ ist so aufgebaut, dass $(a_{m+1}^k, b_1^{k-1}) \in S^2$ und $(b_{m+1}^k, a_1^{k-1}) \in S^2$ gilt. Aus Behauptung 5.7 folgen also $\mathcal{M}^2, a_{m+1}^k \not\models \beta_1^{k-1}$ und $\mathcal{M}^2, b_{m+1}^k \not\models \alpha_1^{k-1}$. Wiederum gilt nach Konstruktion für alle $u \in W$ auch $(u, a_{m+1}^k) \in S^2$ und $(u, b_{m+1}^k) \in S^2$. Damit sind w' und w'' keine Welten aus W, sondern müssen aus W_k^{gen} sein. Da $\mathcal{M}^2, w' \not\models \alpha_i^k$ und $\mathcal{M}^2, w'' \not\models \beta_i^k$ gilt, folgt mit Behauptung 5.7 $w' = a_i^k$ und $w'' = b_i^k$. Aus $(w, a_i^k) \in S^2$, $(w, b_i^k) \in S^2$ und der Konstruktion von S^2 folgt $w \notin \xi(q_i)$ und damit gilt $\mathcal{M}, w \not\models q_i$.
Für den Induktionsschritt sei $\varphi = \gamma \star \delta$ mit $\star \in \{\wedge, \vee, \twoheadrightarrow\}$. Wir zeigen

$$\mathcal{M}^2, w \models (\gamma \star \delta)^2 \iff \mathcal{M}, w \models \gamma \star \delta \ .$$

Man beachte hierbei, dass $(\gamma \star \delta)^2 = \gamma^2 \star \delta^2$ gilt. Die Fälle $\star = \wedge$ und $\star = \vee$ folgen direkt aus der Definition von $\models$ und der Induktionsvoraussetzung.
Wir betrachten jetzt den Fall $\star = \twoheadrightarrow$ und gehen zuerst von $\mathcal{M}, w \not\models \varphi$ aus ($\varphi = \gamma \twoheadrightarrow \delta$). Dann gibt es eine Welt $w' \in W$ mit $(w, w') \in S$, für die $\mathcal{M}, w' \models \gamma$ und $\mathcal{M}, w' \not\models \delta$ gilt. Nach Induktionsvoraussetzung gilt dies dann auch für die übersetzten Formeln in $\mathcal{M}^2$, also $\mathcal{M}^2, w' \models \gamma^2$ und $\mathcal{M}^2, w' \not\models \delta^2$ und es folgt $\mathcal{M}^2, w \not\models \varphi^2$.
Für die andere Richtung sei $w \in W$ und $\mathcal{M}^2, w \not\models \varphi^2$. Es muss demnach eine Welt $w' \in W^2$ mit $(w, w') \in S^2$ geben, so dass $\mathcal{M}^2, w' \models \gamma$ und $\mathcal{M}^2, w' \not\models \delta$ gilt. Die Disjunktionen $\alpha_1^k \vee \beta_1^k, \alpha_2^k \vee \beta_2^k, \ldots, \alpha_m^k \vee \beta_m^k$ sind in allen Welten aller Level aus $\mathcal{M}_k^{gen}$ erfüllt. Die Welten der Level $1, 2, \ldots, k-1$ erfüllen jede Ersetzungsformel aus Level k und die Welten des k-ten Levels erfüllen nach Behauptung 5.7 immer nur genau eine Ersetzungsformel aus Level k nicht. Jedes Vorkommen einer Variablen q_i in δ wurde in δ^2 durch $\alpha_i^k \vee \beta_i^k$ ersetzt. Soll δ^2 in einer Welt aus W^2 nicht gelten, so muss diese Welt aus W sein – also $w' \in W$. Damit können wir die Induktionsvoraussetzung anwenden und $\mathcal{M}, w' \models \gamma$ und $\mathcal{M}, w' \not\models \delta$ folgern. Aus $(w, w') \in S^2$ und $w' \in W$ folgt $(w, w') \in S$ und damit gilt auch $\mathcal{M}, w \not\models \varphi$. ▪
Als Reduktionsfunktion für die Reduktion IPL[$\twoheadrightarrow$]-FA $\leq_{\mathrm{m}}^{\log}$ KC[$\wedge, \vee, \twoheadrightarrow, 2$]-FA ergibt sich damit $\langle \varphi, \mathcal{M}, w \rangle \longmapsto \langle \varphi^2, \mathcal{M}^2, w \rangle$, wobei $\langle \varphi, \mathcal{M}, w \rangle$ eine IPL[$\twoheadrightarrow$]-FA-Instanz ist. Behauptung 5.8 zeigt

$$\mathcal{M}, w \models \varphi \iff \mathcal{M}^2, w \models \varphi^2 \ .$$

Die Reduktion ist offensichtlich in logarithmischem Platz berechenbar und damit gilt nach Bemerkung 5.2, dass KC[$\wedge, \vee, \twoheadrightarrow, 2$]-FA P-hart ist. □

Da jede KC[$\wedge, \vee, \twoheadrightarrow, 2$]-FA-Instanz auch eine IPL[$\wedge, \vee, \twoheadrightarrow, 2$]-FA-Instanz ist, lässt sich Theorem 5.6 direkt auf IPL und BPL übertragen.

Bemerkung 5.7 *Die Probleme* IPL[$\wedge, \vee, \rightarrow, 2$]-FA *und* BPL[$\wedge, \vee, \rightarrow, 2$]-FA *sind* P*-hart.*

5.2 Optimalität bezüglich der Variablenzahl

Wir haben im vorherigen Abschnitt für einige Fragmente gezeigt, dass ihr Formelauswertungsproblem P-hart ist. Uns interessiert jetzt die Zahl der Variablen, die man mindestens benötigt, um in diesen Fragmenten die P-Härte zu erreichen. Die Logiken KC ohne und mit einer Variablen haben ein NC^1-vollständiges Formelauswertungsproblem (Theoreme 2.31 und 4.30), somit ist das Resultat aus Theorem 5.6 bezüglich der Variablenzahl optimal. Auch für IPL ist das Formelauswertungsproblem mit weniger als zwei Variablen nicht mehr P-hart (Theorem 4.17) und unser Resultat (Bemerkung 5.7) ist optimal. In Theorem 5.3 zeigten wir, dass FPL mit einer Variablen ein P-hartes Formelauswertungsproblem hat. Dies gilt auch für das Implikationsfragment von FPL. Im Folgenden wollen wir zeigen, dass die Komplexität von FPL[0]-FA unterhalb von P liegt. Dabei nutzen wir, dass es für FPL[0]-Formeln ähnlich wie für IPL[1]-Formeln eine systematische Beschreibung von Repräsentanten der Formeläquivalenzklassen gibt. Diese stammt von Visser [Vis80].

Theorem 5.8 *Das Problem für* FPL[0]-FA *ist in* LOGCFL*.*

Beweis. Visser [Vis80] zeigte 1980, dass es unendlich viele Äquivalenzklassen von FPL[0]-Formeln bezüglich der FPL[0]-Modelle gibt. Außerdem gibt er eine systematische Konstruktion von Repräsentanten dieser Klassen an. Die Konstruktion ermöglicht es uns, jede Formel durch eine kurze Zeichenkette zu repräsentieren. Die Zeichenkette ist der Index des Repräsentanten und wir bezeichnen ihn als *Visser-Index*. Den Repräsentanten selbst nennen wir *Visser-Formel*. Hier besteht eine gewisse Ähnlichkeit zu den Rieger-Nishimura-Indizes. Allerdings war deren Länge loglog in der Länge der entsprechenden Rieger-Nishimura-Formel, während der Visser-Index nur logarithmisch lang in der Länge der zugehörigen Visser-Formel ist. Da die FPL[0]-Modelle alle irreflexiv sind, ist die Länge des längsten Pfades, der in einer Welt startet, eindeutig bestimmt. Wir nennen diese Länge *Pfadlänge einer Welt* und zeigen, dass sie die Welt eindeutig charakterisiert. Als Zusammenhang zwischen dieser Länge und den Visser-Indizes zeigen wir, dass eine Formel genau dann in einer Welt erfüllt ist, wenn ihr Visser-Index größer als die Pfadlänge der Welt ist. Damit können wir einen LOGCFL-Algorithmus angeben, der das Formelauswertungsproblem für FPL[0] entscheidet. Da es in FPL[0] keine Variablen gibt, betrachten wir die Modelle nur als Paar bestehend aus der Menge der Welten und der Sichtbarkeitsrelation.
Wir geben jetzt die Visser-Formeln an (siehe [Vis80, Definition 4.3]). Es sei $i \in \mathbb{N} \cup \{\omega\}$, wobei $\omega > j$ für alle $j \in \mathbb{N}$ gilt. Dann sind

$$\alpha_0 := \bot, \qquad \alpha_\omega := \top, \qquad \alpha_{i+1} := \top \rightarrow \alpha_i \quad \text{für } i \in \mathbb{N}$$

die Visser-Formeln. Für eine Visser-Formel α_i mit $i \in \mathbb{N} \cup \{\omega\}$ ist i der Visser-Index.

Behauptung 5.9 ([Vis80, Facts 4.4(iii)])

(1) *Die Visser-Formeln sind alle paarweise nicht* $\mathfrak{K}^{\mathsf{i}}_{irr}[0]$*-äquivalent*[2]*.*

(2) *Jede* FPL[0]*-Formel ist zu genau einer Visser-Formel* $\mathfrak{K}^{\mathsf{i}}_{irr}[0]$*-äquivalent.*

Beweis der Behauptung. Visser [Vis80, Facts 4.4] liefert einen Ansatz, wie man diese Behauptung syntaktisch mittels natürlichen Schließens beweisen kann. Wir führen den Beweis mit Hilfe der Semantik der Kripke-Modelle.
Wir zeigen (1), indem wir Modelle angeben, mit denen sich die Visser-Formeln unterscheiden lassen. Wir definieren die Modelle

$$\mathcal{M}_i \quad := \quad (\{1, 2, \ldots, i\}, >)$$

für alle $i > 0$ und zeigen jetzt für $0 < j \leq i$ und $k \in \mathbb{N} \cup \{\omega\}$

$$\mathcal{M}_i, j \models \alpha_k \quad \Longleftrightarrow \quad k \geq j$$

mit vollständiger Induktion über k. Der Induktionsanfang für $k = 0$ ist trivial, da α_0 nirgends erfüllt wird. Auch der Fall $k = \omega$ ist klar. Den Induktionsschritt zeigen wir mit folgenden Äquivalenzen:

$$\begin{aligned}
&\mathcal{M}_i, j \models \alpha_k \quad (= \top \rightarrowtriangle \alpha_{k-1}) && \text{(i)}\\
&\Leftrightarrow \quad \forall x < j : \mathcal{M}_i, x \models \alpha_{k-1} && \text{(ii)}\\
&\Leftrightarrow \quad \forall x < j : k - 1 \geq x && \text{(iii)}\\
&\Leftrightarrow \quad k \geq j\ . && \text{(iv)}
\end{aligned}$$

Die erste Äquivalenz folgt aus der Interpretation von $\rightarrowtriangle$, die zweite ist die Anwendung der Induktionsvoraussetzung und die dritte ist offensichtlich. Damit folgt jetzt für alle $i > 0$

$$\mathcal{M}_i, i \models \alpha_i \quad \text{und} \quad \mathcal{M}_i, i \not\models \alpha_{i-1}\ . \quad (*)$$

Weil außerdem α_ω zu keinem anderen α_i äquivalent ist, gilt (1).
Für (2) benötigen wir folgende Hilfsaussage: Für alle $i, j \in \mathbb{N} \cup \{\omega\}$ gilt

$$i \leq j \iff \forall \mathcal{M} = (W, S) \in \mathfrak{K}^{\mathsf{i}}_{irr}[0], \forall w \in W : \mathcal{M}, w \models \alpha_i \Rightarrow \mathcal{M}, w \models \alpha_j\ . \ (**)$$

Wir zeigen jetzt die Richtung „⇒“ von (∗∗) mit vollständiger Induktion über die Differenz von i und j. Der Fall $j = \omega$ ist klar und auch der Induktionsanfang für $i = j$ ist trivial. Für den Induktionsschritt gelte jetzt $j = i + k$ für ein $k > 0$. Außerdem seien $\mathcal{M} = (W, S)$ ein beliebiges FPL[0]-Modell und $w \in W$ eine Welt. Dann gelten folgende Äquivalenzen:

[2] $\mathrm{FPL}[0] = (\mathfrak{F}^{\mathsf{i}}[0], \mathfrak{K}^{\mathsf{i}}_{irr}[0])$

$$
\begin{aligned}
&\mathcal{M}, w \models \alpha_i && \text{(i)}\\
\Rightarrow\quad &\mathcal{M}, w \models \alpha_{j-1} && \text{(ii)}\\
\Rightarrow\quad &\forall v \in W, (w,v) \in S : \mathcal{M}, v \models \alpha_{j-1} && \text{(iii)}\\
\Rightarrow\quad &\mathcal{M}, w \models \top \twoheadrightarrow \alpha_{j-1} \quad (= \alpha_j)\ . && \text{(iv)}
\end{aligned}
$$

Die erste Folgerung gilt nach Induktionsvoraussetzung ($i \leq j-1$), die zweite folgt aus der Monotonie und die dritte aus der Interpretation von $\twoheadrightarrow$. Die andere Richtung „$\Leftarrow$“ von $(**)$ zeigen wir mit der Kontraposition. Sei $\mathcal{M}_i$ wie im Beweis von (1) definiert. Aus $(*)$ folgt für $i > j$, dass $\mathcal{M}_i, i \models \alpha_i$ und $\mathcal{M}_i, i \not\models \alpha_j$ gilt. Damit ist die Hilfsaussage $(**)$ bewiesen.

Nun wenden wir uns der eigentlichen Aussage (2) zu. Sei φ eine beliebige FPL[0]-Formel. Wir zeigen die Behauptung mit vollständiger Induktion über den Aufbau von φ. Der Induktionsanfang ist trivial, weil aus $\varphi = \bot$ direkt $\varphi \equiv_{\mathfrak{K}^{\mathsf{i}}_{irr}[0]} \alpha_0$ folgt. Für den Induktionsschritt sei $\varphi = \beta \star \gamma$ mit $\star \in \{\wedge, \vee, \twoheadrightarrow\}$. Nach Induktionsvoraussetzung gibt es $j, k \in \mathbb{N} \cup \{\omega\}$ mit $\beta \equiv_{\mathfrak{K}^{\mathsf{i}}_{irr}[0]} \alpha_j$ und $\gamma \equiv_{\mathfrak{K}^{\mathsf{i}}_{irr}[0]} \alpha_k$.

Als ersten Fall betrachten wir $\star = \wedge$. Es seien $\mathcal{M} = (W, S)$ ein beliebiges FPL[0]-Modell und $w \in W$ eine Welt aus $\mathcal{M}$, dann gelten folgenden Äquivalenzen:

$$
\begin{aligned}
&\mathcal{M}, w \models \varphi \quad (= \beta \wedge \gamma) && \text{(i)}\\
\Leftrightarrow\quad &\mathcal{M}, w \models \alpha_j \wedge \alpha_k && \text{(ii)}\\
\Leftrightarrow\quad &\mathcal{M}, w \models \alpha_j \text{ und } \mathcal{M}, w \models \alpha_k && \text{(iii)}\\
\Leftrightarrow\quad &\mathcal{M}, w \models \alpha_{\min\{j,k\}}\ . && \text{(iv)}
\end{aligned}
$$

Die erste Äquivalenz folgt aus der Induktionsvoraussetzung, die zweite aus der Interpretation von $\wedge$ und wegen $(**)$ sind (iii) und (iv) äquivalent.

Im Fall $\star = \vee$ kann man analog $\varphi \equiv_{\mathfrak{K}^{\mathsf{i}}_{irr}[0]} \alpha_{\max\{j,k\}}$ zeigen.

Den Fall $\star = \twoheadrightarrow$ unterteilen wir in $j \leq k$ und $j > k$. Wieder seien $\mathcal{M} = (W, S) \in \mathfrak{K}^{\mathsf{i}}_{irr}[0]$ und $w \in W$. Für $j \leq k$ ergeben sich die folgenden Äquivalenzen:

$$
\begin{aligned}
&\mathcal{M}, w \models \varphi \quad (= \beta \twoheadrightarrow \gamma) && \text{(i)}\\
\Leftrightarrow\quad &\mathcal{M}, w \models \alpha_j \twoheadrightarrow \alpha_k && \text{(ii)}\\
\Leftrightarrow\quad &\forall v \in W, (w,v) \in S : \mathcal{M}, v \models \alpha_j \Rightarrow \mathcal{M}, v \models \alpha_k && \text{(iii)}\\
\Leftrightarrow\quad &\mathcal{M}, w \models \alpha_\omega\ . && \text{(iv)}
\end{aligned}
$$

Die Äquivalenzen zwischen (i) und (ii) bzw. (ii) und (iii) sind klar. Die Aussage $\mathcal{M}, v \models \alpha_j \Rightarrow \mathcal{M}, v \models \alpha_k$ gilt unabhängig von der Wahl des Modells und der Welt immer nach $(**)$, weil wir $j \leq k$ vorausgesetzt haben und damit sind auch (iii) und (iv) äquivalent.

Für $j > k$ gelten folgende Äquivalenzen:

$$
\begin{aligned}
&\mathcal{M}, w \models \varphi \quad (= \beta \twoheadrightarrow \gamma) && \text{(i)}\\
&\Leftrightarrow \quad \mathcal{M}, w \models \alpha_j \twoheadrightarrow \alpha_k && \text{(ii)}\\
&\Leftrightarrow \quad \forall v \in W, (w,v) \in S : \mathcal{M}, v \models \alpha_j \Rightarrow \mathcal{M}, v \models \alpha_k && \text{(iii)}\\
&\Leftrightarrow \quad \forall v \in W, (w,v) \in S : \mathcal{M}, v \models \alpha_k && \text{(iv)}\\
&\Leftrightarrow \quad \mathcal{M}, w \models \top \twoheadrightarrow \alpha_k \quad (= \alpha_{k+1}) \ . && \text{(v)}
\end{aligned}
$$

Alle Äquivalenzen außer der zwischen (iii) und (iv) sind klar. Dass (iii) aus (iv) folgt, ist ebenfalls offensichtlich. Die andere Richtung folgt indirekt aus $(**)$. Gäbe es eine Welt v in $\mathcal{M}$ mit $(w,v) \in S$ und $\mathcal{M}, v \not\models \alpha_k$, so gäbe es auch eine Welt $v' \in W$ mit $(w,v') \in S$ und $\mathcal{M}, v' \models \alpha_j$ und $\mathcal{M}, v' \not\models \alpha_k$. Mit Hilfe der Modelle $\mathcal{M}$ aus dem Beweis von (1) lässt sich leicht eine solche Situation konstruieren.
Damit ist gezeigt, dass jede FPL[0]-Formel zu genau einer Visser-Formel äquivlant ist. ▪

Wir verwenden den Begriff *Visser-Index* jetzt nicht nur für Visser-Formeln, sondern erweitern ihn für alle FPL[0]-Formeln. Eine Formel φ hat den Visser-Index i, wenn $\varphi \equiv_{\mathfrak{K}^{\mathsf{i}}_{irr}[0]} \alpha_i$ gilt. Im weiteren Verlauf interessiert uns, wie aufwändig es ist, den Visser-Index einer Formel zu bestimmen. Dafür definieren wir das folgende Problem:

Problem:	ÄQV-FORMEL
Eingabe:	$\langle \varphi, i \rangle$, wobei $\varphi \in \mathfrak{F}^{\mathsf{i}}[0]$ und $i \in \mathbb{N} \cup \{\omega\}$.
Frage:	Gilt $\varphi \equiv_{\mathfrak{K}^{\mathsf{i}}_{irr}[0]} \alpha_i$?

Behauptung 5.10 *Das Problem* ÄQV-FORMEL *ist in* LOGdetCFL.

Beweis der Behauptung. Aus dem Beweis der Behauptung 5.9(2) ergibt sich direkt die folgende Fallunterscheidung:

$$
\begin{array}{lll}
\text{Wenn } \varphi = \bot, & \text{dann} & \varphi \equiv_{\mathfrak{K}^{\mathsf{i}}_{irr}[0]} \alpha_0.\\
\text{Wenn } \varphi \equiv_{\mathfrak{K}^{\mathsf{i}}_{irr}[0]} \alpha_a \wedge \alpha_b, & \text{dann} & \varphi \equiv_{\mathfrak{K}^{\mathsf{i}}_{irr}[0]} \alpha_{\min\{a,b\}}.\\
\text{Wenn } \varphi \equiv_{\mathfrak{K}^{\mathsf{i}}_{irr}[0]} \alpha_a \vee \alpha_b, & \text{dann} & \varphi \equiv_{\mathfrak{K}^{\mathsf{i}}_{irr}[0]} \alpha_{\max\{a,b\}}.\\
\text{Wenn } \varphi \equiv_{\mathfrak{K}^{\mathsf{i}}_{irr}[0]} \alpha_a \twoheadrightarrow \alpha_b, & \text{dann} & \begin{cases} \varphi \equiv_{\mathfrak{K}^{\mathsf{i}}_{irr}[0]} \alpha_\omega & \text{wenn } a \leq b\\ \varphi \equiv_{\mathfrak{K}^{\mathsf{i}}_{irr}[0]} \alpha_{b+1} & \text{wenn } a > b. \end{cases}
\end{array}
$$

Basierend auf dieser Fallunterscheidung entscheidet Algorithmus 8 ÄQV-FORMEL. Die Korrektheit folgt direkt aus der Fallunterscheidung.
Jede Variable in Algorithmus 8 kann in logarithmischem Platz gespeichert werden, denn aus $\varphi \equiv_{\mathfrak{K}^{\mathsf{i}}_{irr}[0]} \alpha_i$ folgt $i \leq |\varphi|$ oder $i = \omega$. Der Algorithmus durchläuft die Formel rekursiv und berechnet den Visser-Index jeder Teilformel genau einmal, also ist die Laufzeit polynomiell. Sämtliche Informationen, die für die Organisation der

Algorithmus 8 `Visser-Index-Tester`

Eingabe: Formel $\varphi \in \mathfrak{F}^{\mathrm{i}}[0]$, Index $i \in \mathbb{N} \cup \{\omega\}$

1: **wenn** $\texttt{VIndex}(\varphi) = i$ **dann** akzeptieren **sonst** lehne ab

2: **Funktion** $\texttt{VIndex}(\psi)$ // liefert Visser-Index zurück
3: **wenn** $\psi = \bot$ **dann Rückgabe** 0
4: **wenn** $\psi = \beta \wedge \gamma$ **dann Rückgabe** $\min\{\texttt{VIndex}(\beta), \texttt{VIndex}(\gamma)\}$
5: **wenn** $\psi = \beta \vee \gamma$ **dann Rückgabe** $\max\{\texttt{VIndex}(\beta), \texttt{VIndex}(\gamma)\}$
6: **wenn** $\psi = \beta \to \gamma$ **dann**
7: $\quad b := \texttt{VIndex}(\beta)$
8: $\quad c := \texttt{VIndex}(\gamma)$
9: $\quad$ **wenn** $b \leq c$ **dann Rückgabe** ω
10: $\quad\quad$ **sonst Rückgabe** $c + 1$

Rekursion benötigt werden, können auf einem Stapel gespeichert werden. Offensichtlich ist der Algorithmus deterministisch und kann auf einer Turingmaschine implementiert werden, die polynomielle Laufzeit, logarithmischen Speicherplatz und zusätzlich einen Stapel zur Verfügung hat. Mit Theorem 2.14 folgt also, dass ÄqV-Formel in LOGdetCFL ist. ▪

Im nächsten Abschnitt dieses Beweises zeigen wir, dass für die Formelauswertung in einer Welt eines FPL[0]-Modells einzig die Länge des längsten in dieser Welt beginnenden Pfades entscheidend ist. Es sei $\mathcal{M} = (W, S)$ ein FPL[0]-Modell. Wir definieren für $\mathcal{M}$ jetzt eine Funktion $lp_{\mathcal{M}} : W \mapsto \mathbb{N}$, die eine Welt w auf diese Länge abbildet:

$$lp_{\mathcal{M}}(w) \quad := \quad \begin{cases} 1, & \text{falls} \quad \nexists v \in W \text{ mit } (w,v) \in S \\ \max\limits_{(w,v)\in S}\{lp_{\mathcal{M}}(v)\} + 1, & \text{sonst .} \end{cases}$$

Behauptung 5.11 *Sei $\mathcal{M} = (W, S)$ ein* FPL[0]*-Modell. Für jede Visser-Formel α_i und jede Welt $w \in W$ gilt*

$$\mathcal{M}, w \models \alpha_i \quad \Longleftrightarrow \quad lp_{\mathcal{M}}(w) \leq i \ .$$

Beweis der Behauptung. Wir beweisen dies mit vollständiger Induktion über den Visser-Index i. Die Fälle $i = 0$ und $i = \omega$ sind klar. Für den Induktionsschritt sei $i \in \mathbb{N} \setminus \{0\}$, dann gelten die folgenden Äquivalenzen:

$$\begin{aligned} &\mathcal{M}, w \models \alpha_i && \text{(i)} \\ \Leftrightarrow \quad &\forall v \in W, (w,v) \in S : \mathcal{M}, v \models \alpha_{i-1} && \text{(ii)} \\ \Leftrightarrow \quad &\forall v \in W, (w,v) \in S : lp_{\mathcal{M}}(v) \leq i - 1 && \text{(iii)} \\ \Leftrightarrow \quad &lp_{\mathcal{M}}(w) \leq i \ . && \text{(iv)} \end{aligned}$$

Die Interpretation von $\twoheadrightarrow$ liefert die erste Äquivalenz und aus der Induktionsvoraussetzung folgt die zweite Äquivalenz. Da S irreflexiv ist, gilt $w \neq v$ und es folgt, dass (iii) und (iv) äquivalent sind. ▪
Algorithmus 9 entscheidet FPL[0]-FA mit den Ressourcen von LOGCFL.

Algorithmus 9 `Formelauswertung für` FPL[0]

Eingabe: Formel $\varphi \in \mathfrak{F}^{\mathsf{i}}[0]$, Modell $\mathcal{M} \in \mathfrak{K}^{\mathsf{i}}_{irr}[0]$, Welt w aus $\mathcal{M}$

1: rate nichtdeterministisch einen Visser-Index $i \in \{0, 1, \ldots, |\varphi|\} \cup \{\omega\}$
2: **wenn** $(\varphi, i) \in$ ÄQV-FORMEL **dann**
3: rate nichtdeterministisch eine natürliche Zahl $n \leq i$
4: **wenn** $lp_{\mathcal{M}}(w) = n$ **dann** akzeptiere **sonst** lehne ab
5: **sonst** lehne ab

In den ersten beiden Schritten wird der Visser-Index der eingegebenen Formel bestimmt. Aus Behauptung 5.10 folgt, dass diese Schritte mit den Ressourcen von LOGCFL ausgeführt werden können.[3] In den nächsten beiden Schritten wird die Länge des längsten Pfades, der in w beginnt, geraten und verifiziert. Da es sich bei FPL-Modellen um kreisfreie und gerichtete Graphen handelt, ist die Frage nach dem längsten Pfad genau das LPFAD-Problem von dem die NL-Vollständigkeit bekannt ist [JT07]. Somit kann die Verifikation mit den Ressourcen von NL ausgeführt werden.[4] Die Korrektheit der Entscheidung in Schritt 4 folgt aus den Behauptungen 5.9 und 5.11(1). Insgesamt kann Algorithmus 9 also auf einer nichtdeterministischen Turingmaschine mit polynomieller Laufzeit, logarithmischem Speicherplatz und einen zusätzlichen Stapel implementiert werden. Nach Theorem 2.14 sind dies die Ressourcen von LOGCFL. □

Durch eine leichte Modifikation von Algorithmus 9 lässt sich die Formelauswertung für FPL[0] in zwei getrennte Schritte zerlegen. Im ersten Schritt bestimmt man den Visser-Index der Formel in LOGdetCFL. Im zweiten wird geprüft, ob die Länge des längsten Pfades beginnend in der gegebenen Welt kleiner als der Visser-Index ist, was in NL möglich ist. LOGdetCFL und NL sind in LOGCFL enthalten. Unklar ist, ob sie echt enthalten oder eine der beiden Klassen gleich LOGCFL ist, was aber als unwahrscheinlich gilt. Da NL und LOGdetCFL nicht vergleichbar sind, geben wir hier LOGCFL als obere Schranke von FPL[0]-FA an. Ob FPL[0]-FA auch LOGCFL-hart ist, bleibt offen. Allerdings ist es mit dieser Teilung des Algorithmus auch möglich, das Problem von einem LOGdetCFL-Algorithmus lösen zu lassen, der am Ende eine Frage an ein NL-Orakel stellt. Genauso ist ein NL-Algorithmus möglich, der zum Schluss eine Frage an ein LOGdetCFL-Orakel stellt. Diese beiden Varianten legen nahe, dass FPL[0]-FA möglicherweise nicht LOGCFL-hart ist.

Theorem 5.9 *Das Problem* FPL$[\bot, \twoheadrightarrow, 0]$-FA *ist* NL*-hart.*

[3] Stattdessen könnte man auch alle Möglichkeiten iterativ durchprobieren, da der Visser-Index von φ aus $i \in \{0, 1, \ldots, |\varphi|\} \cup \{\omega\}$ ist.
[4] Auch hier müsste die Länge nicht geraten werden.

Beweis. Aus Behauptung 5.11 kann man ableiten, dass man mit den Visser-Formeln die Tiefe eines Modells bestimmen kann. In diesem Sinne kann man die Visser-Formeln verwenden, um die Länge des längsten Pfades zu „berechnen“. Wir geben jetzt eine Reduktion von LPFAD auf FPL[$\bot, \rightarrow, 0$]-FA an.
Sei $\langle G = (V, E), v \in V, n \in \mathbb{N}\rangle$ eine LPFAD-Instanz. Da G nach Voraussetzung kreisfrei und gerichtet ist, kann der Graph direkt als FPL[$\bot, \rightarrow, 0$]-Modell aufgefasst werden. Nach Behauptung 5.11 ist die Visser-Formel α_i genau dann in v erfüllt, wenn $lp_G(v) \leq i$ gilt. Der längste Pfad von v aus hat also genau dann die Länge n, wenn $G, v \models \alpha_i$ und $G, v \not\models \alpha_{i-1}$ gilt. Da NL unter Komplementierung abgeschlossen ist, folgt die Korrektheit der Reduktion. Die NL-Vollständigkeit von LPFAD ist in [JT07] gezeigt und es folgt, dass FPL[$\bot, \rightarrow, 0$]-FA NL-hart ist. □

Diese untere Schranke überträgt sich direkt auf FPL[$\rightarrow, n$]-FA für $n > 0$, da dort eine Variable die Rolle von $\bot$ einnehmen kann. Sie gilt ebenfalls für entsprechenden Fragmente von BPL. Die exakte Komplexität von FPL[0]-FA bleibt offen.

Bemerkung 5.10 *Die Probleme* BPL[$\bot, \rightarrow, 0$]-FA *und für* $n > 0$ BPL[$\rightarrow, n$]-FA *und* FPL[$\rightarrow, n$]-FA *sind* NL*-hart.*

5.3 Zusammenfassung

Wir haben in diesem Kapitel für viele intuitionistische Logiken die P-Härte des Formelauswertungsproblems gezeigt. Damit erreicht das Problem bei diesen Fragmenten die höchste mögliche Komplexität, denn P als obere Schranke für das Formelauswertungsproblem aller Fragmente ist aus Theorem 2.30 bekannt.
Die Resultate aus Abschnitt 5.1 können wir wie folgt zusammenfassen.

Theorem 5.11 *Folgende Probleme sind* P*-vollständig:*

(1) L[O]-FA *für* L $\in \{\mathrm{KC}, \mathrm{IPL}, \mathrm{BPL}\}$ *und* $\{\rightarrow\} \subseteq O \subseteq \{\bot, \wedge, \vee, \rightarrow\}$,

(2) L[O, n]-FA *für* L $\in \{\mathrm{FPL}, \mathrm{BPL}\}$, $\{\bot, \rightarrow\} \subseteq O \subseteq \{\bot, \wedge, \vee, \rightarrow\}$ *und* $n \in \mathbb{N}^+ \cup \{\omega\}$,

(3) BPL[O, n]-FA *für* $\{\bot, \vee, \rightarrow\} \subseteq O \subseteq \{\bot, \wedge, \vee, \rightarrow\}$ *und* $n \in \mathbb{N} \cup \{\omega\}$ *und*

(4) L[O, n]-FA *für* L $\in \{\mathrm{KC}, \mathrm{IPL}, \mathrm{BPL}\}$, $\{\wedge, \vee, \rightarrow\} \subseteq O \subseteq \{\bot, \wedge, \vee, \rightarrow\}$ *und* $n \in \{2, 3, \dots\} \cup \{\omega\}$.

Beweis. Die obere Schranke folgt aus Theorem 2.30. Die P-Härte folgt für (1) aus Theorem 5.1 und Bemerkung 5.2, für (2) aus Theorem 5.3 und Bemerkung 5.4, für (3) aus Theorem 5.5 und für (4) aus Theorem 5.6 und Bemerkung 5.7. □

In Abschnitt 5.2 untersuchten wir, wie viele Variablen in einem Fragment notwendig sind, damit dessen Formelauswertungsproblem P-hart ist. Teilweise stammten die Ergebnisse dazu bereits aus vorherigen Kapiteln.

Theorem 5.12 *Es gilt*

$$\mathrm{L}[n]\text{-FA } \textit{ist } \mathsf{P}\textit{-hart} \iff \begin{cases} n \geq 0, & \textit{falls} \quad \mathrm{L} = \mathrm{BPL} \\ n \geq 1, & \textit{falls} \quad \mathrm{L} = \mathrm{FPL} \\ n \geq 2, & \textit{falls} \quad \mathrm{L} \in \{\mathrm{KC}, \mathrm{IPL}\} \\ n = \omega, & \textit{falls} \quad \mathrm{L} \in \{\mathrm{KC}[\twoheadrightarrow], \mathrm{IPL}[\twoheadrightarrow]\} \ . \end{cases}$$

Beweis. In Theorem 5.8 zeigten wir, dass das Formelauswertungsproblem für FPL[0] in LOGCFL liegt. Für IPL[1] ist die Formelauswertung nach Theorem 4.17 in AC^1, für KC[1] in NC^1 (Theorem 4.23). Die Implikationsfragmente von IPL mit endlich vielen Variablen sind endlich erzeugt [Sko53, McK68] und damit ist die Formelauswertung dort in NC^1 möglich (Theorem 3.3). □

Interessanterweise decken sich die Zahlen der Variablen aus Theorem 5.11 für die Fragmente von BPL, FPL und IPL genau mit der denen der Variablen, die für die PSPACE-Härte der entsprechenden Tautologieprobleme notwendig sind. Rybakov [Ryb06] zeigte, dass BPL[0]-TAUT, FPL[1]-TAUT und IPL[2]-TAUT PSPACE-vollständig sind. Wir haben gezeigt, dass IPL[1]-TAUT in LOGdetCFL liegt (Bemerkung 4.9). Für FPL[0] ist das Tautologieproblem ebenfalls in LOGdetCFL, da man nur testen muss, ob eine gegebene Formel äquivalent zu $\top (= \alpha_\omega)$ ist (siehe Behauptung 5.10). Im Gegensatz zu Rybakov können wir uns im Beweis von Theorem 5.3 für FPL[1] auf $\bot$ und $\twoheadrightarrow$ beschränken. Die von ihm verwendeten Formeln enthalten zusätzlich noch $\vee$. Bei KC und IPL ist nur die Verwendung von $\twoheadrightarrow$ für die P-Härte der Formelauswertung notwendig.
Betrachtet man nur die Anzahlen der zugelassenen Variablen und vernachlässigt die verwendeten Operatoren, so lassen sich die Ergebnisse dieses Kapitels wie in Tabelle 5.1 darstellen.

Logik / Variablenzahl	KC[n]	IPL[n]	FPL[n]	BPL[n]
$n = 0$				
$n = 1$				
$n \geq 2$				
$n = \infty$				

nicht P-hart

P-vollständig

Tabelle 5.1: Unsere Komplexitätsresultate für das Formelauswertungsproblem aus Kapitel 5. Für KC[0], KC[1] und IPL[0] liegt es in NC^1, für IPL[1] in AC^1 und für FPL[0] in LOGCFL.

Kapitel 6

Modale Begleiter

In diesem Abschnitt werden wir uns mit einigen Modallogiken beschäftigen. Dabei konzentrieren wir uns auf Logiken, die modale Begleiter der intuitionistischen Logiken aus den letzten Kapiteln sind (Tabelle 2.3). Ziel ist es, die Ergebnisse der intuitionistischen Logiken mit ihren modalen Begleitern zu vergleichen. Generell ist ein modaler Begleiter einer intuitionistischen Logik eine Modallogik, in welche die intuitionistischen Logik mit einer Gödel-Tarski-Übersetzung eingebettet werden kann [Göd32]. Bei einer solchen Einbettung bleibt die Gültigkeit erhalten. Visser [Vis80] gab unter anderem die folgende Einbettung $gt : \mathfrak{F}^{\mathsf{i}} \mapsto \mathfrak{F}$ an:

$$gt(\varphi) \quad := \quad \begin{cases} \bot, & \text{falls} \quad \varphi = \bot \\ \varphi \wedge \Box\varphi, & \text{falls} \quad \varphi \in \mathrm{VAR} \\ gt(\alpha) \wedge gt(\beta), & \text{falls} \quad \varphi = \alpha \wedge \beta \\ gt(\alpha) \vee gt(\beta), & \text{falls} \quad \varphi = \alpha \vee \beta \\ \Box(gt(\alpha) \to gt(\beta)), & \text{falls} \quad \varphi = \alpha \rightarrowtriangle \beta \ . \end{cases}$$

Wenn wir den modalen Begleiter eines Implikationsfragments betrachten, ist leicht zu sehen, dass dieser modale Begleiter selbst kein Implikationsfragment ist.[1] Um zu garantieren, dass auch die übersetzten Formeln nur die strikte Implikation ($\Box(\cdot \to \cdot)$ bzw. $\rightarrowtriangle$) als Operator enthalten, definieren wir gt':

$$gt'(\varphi) \quad := \quad \begin{cases} p, & \text{falls} \quad \varphi \in \mathrm{VAR} \\ gt(\varphi), & \text{sonst} \ . \end{cases}$$

Da es in diesem Kapitel um das Formelauswertungsproblem gehen soll, arbeiten wir mit der Übersetzung gt'. Weil intuitionistische Modelle bereits eine monotone Belegung haben, benötigen wir den Teil $\Box p$ nicht.

Lemma 6.1 *Es Seien $\varphi \in \mathfrak{F}^{\mathsf{i}}$ eine Formel, $\mathcal{M} \in \mathfrak{K}^{\mathsf{i}}$ ein Modell und w eine Welt aus $\mathcal{M}$. Dann gilt*

$$\mathcal{M}, w \models \varphi \iff \mathcal{M}, w \models gt'(\varphi) \ .$$

Für $\varphi \in \mathfrak{F}^{\mathsf{i}}[\rightarrowtriangle]$ gilt $gt'(\varphi) \in \mathfrak{F}[\Box(\cdot \to \cdot)]$.

[1] Ein Implikationsfragment einer Modallogik ist eine Logik, bei der die strikte Implikation der einzige Operator ist.

Dieses Lemma ist offensichtlich, da $\rightarrowtriangle$ durch $\Box(\cdot \rightarrow \cdot)$ definiert ist und gt' in diesem Sinne die Identität ist. In einem modalen Begleiter gibt es mehr Modelle als in der zugehörigen intuitionistischen Logik, über denen eine intuitionistische Formel (bzw. deren Übersetzung) interpretiert werden kann. Daraus ergibt sich, dass die Formelauswertung in einem modalen Begleiter möglicherweise schwerer ist, als in der zugehörigen intuitionistischen Logik. Im Folgenden schreiben wir auch in der Modallogik statt $\Box(\cdot \rightarrow \cdot)$ immer $\rightarrowtriangle$, wenn wir die strikte Implikation meinen.

6.1 Komplexität der Formelauswertung

Die unteren Schranken des Formelauswertungsproblems von intuitionistischen Logiken übertragen sich direkt auf ihre Begleiter, da die intuitionistischen Modelle auch als Modelle in der entsprechenden Modallogik verwendet werden können.

Theorem 6.2 *Die Probleme* S4.2[$\rightarrowtriangle$]-FA*,* PrL[$\bot, \rightarrowtriangle, 1$]-FA*,* K4[$\bot, \vee, \rightarrowtriangle, 0$]-FA *und* S4.2[$\wedge, \vee, \rightarrowtriangle, 2$]-FA *sind* P*-hart.*

Dieses Theorem folgt mit Lemma 6.1 aus den Theoremen 5.1, 5.3, 5.5 und 5.6. Auch die Bemerkungen 5.2, 5.4 und 5.7 lassen sich auf die modalen Begleiter übertragen.

Bemerkung 6.3 *Die Probleme* S4[$\rightarrowtriangle$]-FA*,* K4[$\bot, \rightarrowtriangle, 1$]-FA*,* S4[$\wedge, \vee, \rightarrowtriangle, 2$]-FA *und* K4[$\wedge, \vee, \rightarrowtriangle, 2$]-FA *sind* P*-hart.*

6.1.1 S4.2 und S4 – die modalen Begleiter von KC und IPL

Für den modalen Begleiter S4.2[2] von KC[2] wissen wir, dass das Formelauswertungsproblem P-hart ist. Für die P-Härte bei KC kann man auch keine weitere Variable einsparen, denn KC[1]-FA ist in NC^1 (siehe dazu Abschnitt 4.4). Dagegen können wir bei dem modalen Begleiter S4.2[1] von KC[1] die P-Härte zeigen.

Theorem 6.4 *Das Problem* S4.2[1]-FA *ist* P*-hart.*

Beweis. Wir zeigen Asgep $\leq_{\mathrm{m}}^{\log}$ S4.2[1]-FA. Da Asgep P-hart ist (Theorem 2.21), folgt auch die P-Härte für S4.2[2]-FA.
Es sei $\langle G, s, t\rangle$ eine Asgep-Instanz, wobei $G = (V_\exists \cup V_\forall, E)$ ein Schichtgraph mit m Schichten ist (o.B.d.A. ist m gerade). Weiter ist $V_\exists = V_1 \cup V_3 \cup \cdots \cup V_{m-1}$ und $V_\forall = V_2 \cup V_4 \cup \cdots \cup V_m$. Wie auch schon in vorherigen Beweisen konstruieren wir aus dieser Instanz ein Kripke-Modell $\mathcal{M}_G$ und eine Formel α_G so, dass

$$\langle G, s, t\rangle \in \text{Asgep} \iff \mathcal{M}_G, s \models \alpha_G$$

gilt. Als erstes ergänzen wir zwei Schichten:

$$V_{m+1} := \{u, t_1, t_2\} ,$$
$$V_{m+2} := \{top\} .$$

Im zweiten Schritt werden diese neue Schichten mit Kanten versehen:

$$E' := \{(v, u) \mid v \in V_m\} \cup \{(t, t_1), (t, t_2)\} ,$$
$$E'' := \{(u, top), (t_1, top), (t_2, top)\} ,$$
$$E''' := \{(t_1, t_2), (t_2, t_1)\} .$$

Die Kanten aus E' verbinden jeden Knoten aus Schicht V_m mit u und den Zielknoten t zusätzlich mit t_1 und t_2. Die Kanten aus E'' verbinden die Knoten aus V_{m+1} mit top. In Schicht V_{m+1} werden die Knoten t_1 und t_2 gegenseitig miteinander verbunden. Diese Modifikation ist in einer intuitionistischen Logik nicht möglich, da die Rahmen intuitionistischer Modelle immer kreisfrei sein müssen. Abschließend bilden wir die pseudotransitive Hülle (Definition 2.23) und ergänzen die reflexiven Kanten:

$$E^{trans} := \bigcup_{i=1}^{m} V_i \times V_{\geq i+2} ,$$
$$E^{refl} := \{(v, v) \mid v \in V \cup V_{m+1} \cup V_{m+2}\} .$$

Jetzt können wir den Rahmen $G' = (W, S)$ von $\mathcal{M}_G$ wie folgt angeben:

$$W := V \cup V_{m+1} \cup V_{m+2} ,$$
$$S := E \cup E' \cup E'' \cup E''' \cup E^{trans} \cup E^{refl} .$$

Dieser Graph ist reflexiv, transitiv und jeder Knoten sieht top, also ist G' eine gerichtete Halbordnung und kann als Rahmen für ein S4.2-Modell verwendet werden.

Auch hier ist es wichtig zu bestimmen, aus welcher Schicht eine Welt kommt. Dafür benutzen wir die Belegungsfunktion und „markieren“ die geraden Schichten $V_2, V_4, \ldots, V_{m+2}$ mit der atomaren Aussage p. Aus technischen Gründen wird zusätzlich die Welt t_2 aus Schicht V_{m+1} mit p markiert. Auch dies ist in einem intuitionistischen Modell nicht möglich, da diese Belegung nicht monoton ist. Wir definieren die Belegungsfunktion wie folgt:

$$\xi(a) := V_2 \cup V_4 \cup \cdots \cup V_{m+2} \cup \{t_2\} .$$

Die Konstruktion von $\mathcal{M}_G$ ist jetzt komplett:

$$\mathcal{M}_G := (W, S, \xi, \{p\}) .$$

Ein Beispiel ist in Abbildung 6.1 zu sehen.

Wir konstruieren jetzt die Formel α_G der S4.2[1]-FA-Instanz. Dafür sind noch einige technische Vorbemerkungen notwendig. Es sei

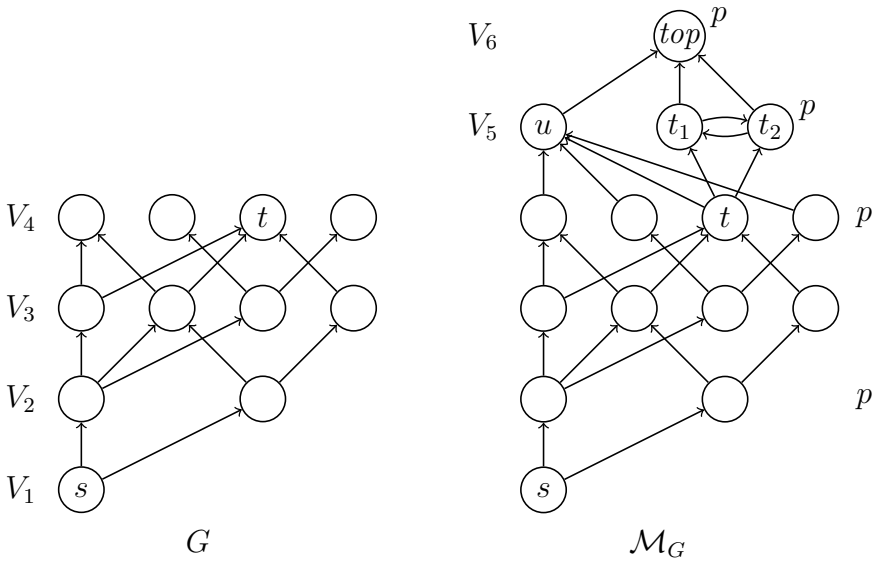

Abbildung 6.1: Links ist eine ASGEP-Instanz $\langle G, s, t\rangle$ abgebildet und rechts das Modell $\mathcal{M}_G$, wie es in Beweis von Theorem 6.4 konstruiert wird. Die Belegungen in den obersten beiden Schichten von $\mathcal{M}_G$ stehen an den Knoten und im unteren Teil rechts neben den Schichten. Reflexive und transitive Kanten in $\mathcal{M}_G$ sind nicht abgebildet.

$$\eta \quad := \quad \neg p \wedge \Diamond(p \wedge \Diamond\neg p) \; .$$

An Hand der Konstruktion von $\mathcal{M}_G$ kann man erkennen, dass t_1 die einzige Welt aus $V_m \cup V_{m+1} \cup V_{m+2}$ ist, die η erfüllt. Die Zielwelt t wiederum ist die einzige Welt aus V_m die t_1 – also eine η erfüllende Welt – als Nachfolger hat. Um zu bestimmen, in welcher Schicht ein Knoten liegt, benutzen wir die im Folgenden definierten Formeln δ_i. Es seien

$$\delta_{m+1} \quad := \quad \Diamond\neg\eta$$

und für $i = m-1, m-2, \ldots, 1$

$$\delta_i \quad := \quad \begin{cases} \Diamond(p \wedge \delta_{i+1}), & \text{falls } i \text{ gerade} \\ \Diamond(\neg p \wedge \delta_{i+1}), & \text{falls } i \text{ ungerade} \; . \end{cases}$$

Wir zeigen, dass δ_i in einer Welt nur dann erfüllt ist, wenn diese Welt aus Schicht V_i oder einer Schicht V_j mit $j < i$ kommt.

Behauptung 6.1 *Seien $w \in V_{\leq m}$ und $i \in \{1, 2, \ldots, m\}$, dann gilt*

$$\mathcal{M}_G, w \models \delta_i \quad \Longleftrightarrow \quad w \in V_{\leq i} \; .$$

Beweis der Behauptung. Wir zeigen die Behauptung mit vollständiger Induktion über $i = m, m-1, \ldots, 1$.
Den Induktionsanfang bildet $i = m$. Es gilt $\mathcal{M}_G, u \models \neg\eta$ und jede Welt aus $V_{\leq m}$ sieht u. Außerdem sieht jede Welt aus $V_{\leq m}$ eine Welt aus V_m und diese erfüllen nach Konstruktion p. Damit erfüllt jede Welt aus $V_{\leq m}$ auch $\Diamond(p \wedge \Diamond\neg\eta)$ $(= \delta_m)$.
Für den Induktionsschritt sei $i < m$. Wir unterscheiden zwischen geradem und ungeradem i und betrachten zunächst den Fall, dass i gerade ist. Es sei $w \in V_{\leq m}$ eine Welt, dann gilt

$$\begin{aligned}
&\mathcal{M}_G, w \models \delta_i \quad (= \Diamond(p \wedge \delta_{i+1})) \\
&\Rightarrow \quad \exists w' \in W, (w, w') \in S : \mathcal{M}_G, w' \models p \text{ und } \mathcal{M}_G, w' \models \delta_{i+1} \\
&\Rightarrow \quad \exists w' \in V_{\leq i+1}, (w, w') \in S : \mathcal{M}_G, w' \models p \text{ und } \mathcal{M}_G, w' \models \delta_{i+1} \\
&\Rightarrow \quad w \in V_{\leq i+1} \ .
\end{aligned}$$

Die zweite Implikation gilt nach Induktionsvoraussetzung, die anderen sind offensichtlich. Da p in keiner Welt aus V_{i+1} erfüllt ist, kann w' nicht aus V_{i+1} sein und es muss $w \in V_{\leq i}$ gelten.
Für die andere Richtung sei w eine Welt aus $V_{\leq i}$. Dann gibt es ein $w' \in V_i$ mit $(w, w') \in S$, für das nach Konstruktion $\mathcal{M}_G, w' \models p$ und nach Induktionsvoraussetzung $\mathcal{M}_G, w' \models \delta_{i+1}$ gelten. Also folgt $\mathcal{M}_G, w \models \delta_i$.
Für den Fall, dass i ungerade ist, verläuft der Beweis analog. ∎
Die Konstruktion von $\mathcal{M}_G$ ist so angelegt, dass t die einzige Welt aus V_m ist, die t_1 sieht, und damit auch die einzige Welt aus dieser Schicht ist, die $\Diamond\eta$ erfüllt. Außerdem lässt sich mit den δ_i-Formeln die obere Schranke der Schichten bestimmen, in denen sich eine Welt befindet. Mit Hilfe dieser beiden Eigenschaften können wir jetzt einen alternierenden Pfad simulieren und konstruieren dafür die Formeln λ_i. Es seien

$$\lambda_m \quad := \quad \Diamond\eta$$

und für $i = m-1, m-2, \ldots, 1$

$$\lambda_i \quad := \quad \begin{cases} \Box((\neg p \wedge \delta_{i+1}) \rightarrow \lambda_{i+1}), & \text{falls } i \text{ gerade} \\ \Diamond(p \wedge \delta_{i+1} \wedge \lambda_{i+1}), & \text{falls } i \text{ ungerade} \ . \end{cases}$$

Diese Formeln bilden die Grundlage um einen alternierenden Pfad durch das Modell zu zeichnen.

Behauptung 6.2 *Es seien $i \in \{1, 2, \ldots, m\}$ und $w \in V_i$, dann gilt*

$$aPfad_G(w, t) \quad \Longleftrightarrow \quad \mathcal{M}_G, w \models \lambda_i \ .$$

Beweis der Behauptung. Auch diese Behauptung zeigen wir mit vollständiger Induktion über $i = m, m-1, \ldots, 1$.

Der Induktionsanfang für $i = m$ wird durch folgende Äquivalenzen gezeigt:

$$\begin{aligned} & aPfad_G(w,t) \\ \Leftrightarrow\quad & w = t \\ \Leftrightarrow\quad & \mathcal{M}_G, w \models \Diamond\eta \quad (= \lambda_m)\ . \end{aligned}$$

Dabei folgt die letzte Äquivalenz direkt aus der Konstruktion von $\mathcal{M}_G$ und η.
Im Induktionsschritt sei $i < m$ und wir unterscheiden wieder zwischen geradem und ungeradem i. Für gerade i ist V_i eine $\forall$-Schicht und es gelten folgende Äquivalenzen für $w \in V_i$:

$$\begin{aligned} & aPfad_G(w,t) && \text{(i)} \\ \Leftrightarrow\quad & \forall w' \in V, (w,w') \in E : aPfad_G(w',t) && \text{(ii)} \\ \Leftrightarrow\quad & \forall w' \in V_{i+1}, (w,w') \in S : \mathcal{M}_G, w' \models \lambda_{i+1} && \text{(iii)} \\ \Leftrightarrow\quad & \forall w' \in W, (w,w') \in S : \mathcal{M}_G, w' \models \neg p \wedge \delta_{i+1} \ \Rightarrow\ \mathcal{M}_G, w' \models \lambda_{i+1} && \text{(iv)} \\ \Leftrightarrow\quad & \mathcal{M}_G, w \models \Box((\neg p \wedge \delta_{i+1}) \rightarrow \lambda_{i+1}) \quad (= \lambda_i)\ . && \text{(v)} \end{aligned}$$

Die erste Äquivalenz folgt direkt aus der Definition von *aPfad* (Definition 2.18) und die zweite aus der Induktionsvoraussetzung. Die Aussagen (iii) und (iv) sind äquivalent, weil nur Welten aus V_{i+1} von w gesehen werden und $\neg p \wedge \delta_{i+1}$ erfüllen. Wichtig hierbei ist, dass p nur in geradzahligen Schichten und δ_{i+1} nur in Welten aus $V_{\leq i+1}$ erfüllt ist (Behauptung 6.1). Die letzte Äquivalenz folgt aus der Definition von $\models$ (Definition 2.5) und $w \in \xi(p)$.
Für i ungerade sind die Äquivalenzen sehr ähnlich:

$$\begin{aligned} & aPfad_G(w,t) && \text{(i)} \\ \Leftrightarrow\quad & \exists w' \in V, (w,w') \in E : aPfad_G(w',t) && \text{(ii)} \\ \Leftrightarrow\quad & \exists w' \in V_{i+1}, (w,w') \in S : \mathcal{M}_G, w' \models \lambda_{i+1} && \text{(iii)} \\ \Leftrightarrow\quad & \exists w' \in W, (w,w') \in S : \mathcal{M}_G, w' \models p \wedge \delta_{i+1} \text{ und } \mathcal{M}_G, w' \models \lambda_{i+1} && \text{(iv)} \\ \Leftrightarrow\quad & \mathcal{M}_G, w \models \Diamond(p \wedge \delta_{i+1} \wedge \lambda_{i+1}) \quad (= \lambda_i)\ . && \text{(v)} \end{aligned}$$

Auch hier ist der entscheidende Punkt die Äquivalenz zwischen (iii) und (iv), die ebenfalls eine Folgerung aus Behauptung 6.1 ist. ∎
Wir setzen jetzt

$$\alpha_G \quad := \quad \lambda_1$$

und mit Behauptung 6.2 folgt

$$\langle G, s, t\rangle \in \textsc{Asgep} \quad \Longleftrightarrow \quad \mathcal{M}_G, s \models \alpha_G\ .$$

Sowohl das Modell $\mathcal{M}_G$ als auch die Formel α_G können aus $\langle G, s, t\rangle$ in logarith-

mischem Platz konstruiert werden und damit gilt ASGEP $\leq_{\mathrm{m}}^{\log}$ S4.2[1]-FA. Mit Theorem 2.21 folgt dann, dass auch S4.2[1]-FA P-hart ist. □

Dieses Resultat lässt sich auch auf S4[1] übertragen, denn jedes S4.2[1]-Modell ist auch ein S4[1]-Modell.

Bemerkung 6.5 *Das Problem* S4[1]-FA *ist* P*-hart.*

Obwohl S4.2 der modale Begleiter von KC (und S4 der von IPL) ist, ist das Formelauswertungsproblem für das Fragment mit nur einer Variablen schwerer als für KC (bzw. IPL). Der entscheidende Unterschied liegt in der Konstruktion der Modelle (siehe Abbildung 6.1). Bei der Modallogik verwenden wir eine Belegungsfunktion, die nicht monoton ist. Außerdem nutzen wir die Negation, um zu überprüfen, ob in einer Welt eine Formel nicht erfüllt ist, ohne dabei auf ihre Nachfolger zu schauen. Diese beiden Möglichkeiten gibt es in den intuitionistischen Logiken nicht.
Die P-Härte von S4[1] und S4.2[1] ist hinsichtlich der Zahl der Variablen optimal.

Theorem 6.6 *Die Probleme* S4[0]-FA *und* S4.2[0]-FA *liegen in* NC^1.

Beweis. Entscheidend für diese Aussage ist, dass man in S4[0]-Formeln die modalen Operatoren $\Box$ (und $\Diamond$) ignorieren kann. Dies beruht im Wesentlichen auf der Reflexivität der S4[0]-Modelle. Eine S4[0]-Formel α ohne modale Operatoren ist entweder gültig oder unerfüllbar. Diese Eigenschaft überträgt sich direkt auf Formeln mit modalen Operatoren. Deswegen gilt für solch ein α und $O \in \{\Box, \Diamond\}$ auch $\alpha \equiv_{\mathfrak{K}_{HO}[0]} O\alpha$ und es ist klar, dass man in einer beliebigen S4[0]-Formel vorkommende modale Operatoren bei einer Auswertung nicht beachten muss.
Sei φ ein beliebige S4[0]-Formel und φ' entsteht aus φ, indem alle modalen Operatoren gestrichen werden. Dann gilt, dass φ in einer beliebigen Welt eines S4[0]-Modells genau dann erfüllbar ist, wenn φ' eine aussagenlogische Tautologie ist. Dies wiederum lässt sich in NC^1 überprüfen [Bus87]. □

6.1.2 PrL – der modale Begleiter von FPL

Bei PrL sind die Modelle transitiv und irreflexiv. Damit ist das Formelauswertungsproblem – ähnlich wie bei FPL – nicht auf BFVP reduzierbar. Hauptsächlich liegt die Schwierigkeit darin, dass Formeln, die mit $\Box$ beginnen, in einer Welt erfüllt sind, die keine Nachfolger hat. In Anlehnung an Theorem 5.8 zeigen wir, dass auch PrL[0]-FA unterhalb von P liegt. Damit ist die P-Härte von PrL$[\bot, \rightarrowtriangle, 1]$-FA (Theorem 6.2) hinsichtlich der Zahl der Variablen optimal.

Theorem 6.7 *Das Problem* PrL[0]-FA *ist in* AC^1.

Beweis. Wir zeigen, dass jedes PrL[0]-Modell auf seinen längsten Pfad reduziert werden kann. Da es in PrL[0] keine Variablen und damit auch keine Belegungen gibt, geben wir die Modelle nur als Paar an, bestehend aus der Menge

der Welten und der Sichtbarkeitsrelation. Wir definieren *lineare Modelle*[2] $\mathcal{L}_n := (\{0, 1, \ldots, n\}, >)$ und verwenden die Funktion $lp_{\mathcal{M}}$, um Welten auf die Länge des längsten, bei ihnen beginnenden Pfades abzubilden. Dabei ist die Funktion $lp_{\mathcal{M}}$ wie auch schon im Beweis von Theorem 5.8 definiert:

$$lp_{\mathcal{M}}(w) \quad := \quad \begin{cases} 1, & \text{falls} \quad \nexists v \in W \text{ mit } (w,v) \in S \\ \max\limits_{(w,v)\in S} \{lp_{\mathcal{M}}(v)\} + 1, & \text{sonst .} \end{cases}$$

Reinhardt [Rei11] zeigte kürzlich, dass AC^1 eine obere Schranke für das Formelauswertungsproblem in PrL[0] beschränkt auf lineare Modelle ist.

Behauptung 6.3 *Seien $\mathcal{M} = (W, S)$ ein* PrL[0]*-Modell und $w \in W$. Dann gilt $(\mathcal{M}, w) \equiv_{\mathfrak{F}[0]} (\mathcal{L}_{lp_{\mathcal{M}}(w)}, lp_{\mathcal{M}}(w))$.*

Beweis der Behauptung. Nach Definition 2.7 ist $(\mathcal{M}, w) \equiv_{\mathfrak{F}[0]} (\mathcal{L}_{lp_{\mathcal{M}}(w)}, lp_{\mathcal{M}}(w))$, wenn für alle PrL[0]-Formeln α

$$\mathcal{M}, w \models \alpha \iff \mathcal{L}_{lp_{\mathcal{M}}(w)}, lp_{\mathcal{M}}(w) \models \alpha$$

gilt. Wir zeigen dies mit vollständiger Induktion über den Aufbau von α. Der Induktionsanfang für $\alpha = \bot$ ist klar. Im Induktionsschritt ist der Fall $\alpha = \beta \to \gamma$ ebenfalls offensichtlich. Im Fall $\alpha = \Box\beta$ gelten folgende Äquivalenzen:

$$\begin{aligned}
&\mathcal{M}, w \models \alpha \quad (= \Box\beta) && \text{(i)}\\
\Leftrightarrow\ & \forall v \in W, (w,v) \in S : \mathcal{M}, v \models \beta && \text{(ii)}\\
\Leftrightarrow\ & \forall v \in W, (w,v) \in S : \mathcal{L}_{lp_{\mathcal{M}}(v)}, lp_{\mathcal{M}}(v) \models \beta && \text{(iii)}\\
\Leftrightarrow\ & \forall v \in W, (w,v) \in S : \mathcal{L}_{lp_{\mathcal{M}}(w)}, lp_{\mathcal{M}}(v) \models \beta && \text{(iv)}\\
\Leftrightarrow\ & \mathcal{L}_{lp_{\mathcal{M}}(w)}, lp_{\mathcal{M}}(w) \models \Box\beta \quad (= \alpha) \ . && \text{(v)}
\end{aligned}$$

Die erste Äquivalenz folgt aus der Definition von $\models$ (Definition 2.5) und die zweite aus der Induktionsvoraussetzung. Die Äquivalenz zwischen (iii) und (iv) basiert auf der Tatsache, dass $\mathcal{L}_{lp_{\mathcal{M}}(v)}$ ein Teilmodell von $\mathcal{L}_{lp_{\mathcal{M}}(w)}$ ist, und die letzte ist wieder offensichtlich. ▪

Für eine PrL[0]-Instanz $\langle \alpha, \mathcal{M}, w \rangle$ lässt sich $lp_{\mathcal{M}}(w)$ mit den Ressourcen von NL berechnen [JT07]. Die Entscheidung $\langle \varphi, \mathcal{L}_{lp_{\mathcal{M}}(w)}, lp_{\mathcal{M}}(w) \rangle \in$ PrL[0]-FA kann mit den Ressourcen von AC^1 getroffen werden [Rei11]. Aus Behauptung 6.3 folgt, dass $\langle \varphi, \mathcal{L}_{lp_{\mathcal{M}}(w)}, lp_{\mathcal{M}}(w) \rangle \in$ PrL[0]-FA genau dann gilt, wenn $\langle \alpha, \mathcal{M}, w \rangle \in$ PrL[0]-FA ist. Damit und aus $\mathsf{NL} \subseteq \mathsf{AC}^1$ folgt, dass PrL[0]-FA $\in \mathsf{AC}^1$ ist. □

Es ist ein offenes Problem, ob AC^1 auch die untere Schranke PrL[0]-FA ist. Aus Lemma 6.1 und Theorem 5.9 folgt NL als untere Schranke, sogar für das Fragment, welches die strikte Implikation als einzigen Operator hat.

[2] Ein Rahmen (W, S) ist linear, wenn für zwei Welten $w_1, w_2 \in W$ mit $w_1 \neq w_2$ immer $(w_1, w_2) \in S$ oder $(w_2, Sw_1) \in S$ gilt.

Bemerkung 6.8 *Das Problem* PrL$[\bot, \rightarrow\!\!\!\triangleright, 0]$-FA *ist* NL*-hart.*

Wir können sowohl für PrL als auch für FPL ohne Variablen keine Vollständigkeitsresultate angeben. Die obere Schranke LOGCFL von FPL$[\bot, \rightarrow\!\!\!\triangleright, 0]$-FA lässt sich auch nicht auf PrL übertragen, da für PrL keine systematische Beschreibung der Formeläquivalenzklassen – ähnlich wie bei FPL (siehe Beweis von Theorem 5.8) – bekannt ist. Dafür konnten wir aber zeigen, dass die P-Härte von PrL$[\bot, \rightarrow\!\!\!\triangleright, 1]$-FA (Theorem 6.2) bezüglich der Zahl der Variablen optimal ist.

6.1.3 S5 – der modale Begleiter von AL

Da in der normalen Aussagenlogik AL das Gesetz des ausgeschlossenen Dritten gültig ist, wird sie nicht als intuitionistische Logik im ursprünglichen Sinn angesehen. Trotzdem kann sie als superintuitionistische Logik aufgefasst werden. Syntaktisch entsteht AL aus IPL, indem man das Gesetz des ausgeschlossenen Dritten als Axiom hinzunimmt. Ihr modaler Begleiter ist dann die Logik S5. Die S5-Modelle basieren auf einem Rahmen, der eine totale Ordnung ist, das heißt, jede Welt sieht jede Welt. Der Rahmen aller Kripke-Modelle für AL besteht aus nur einer Welt, die sich selbst sieht (die einzige totale Ordnung, die kreisfrei ist). Wir zeigen, dass das Formelauswertungsproblem für S5 unterhalb von P liegt. Für AL ist es NC^1-vollständig (siehe Theorem 2.31).

Theorem 6.9 *Das Problem* S5-FA *ist in* LOGdetCFL*.*

Beweis. Es sei $\langle \varphi, \mathcal{M} = (W, S, \xi, \mathrm{VAR}), w \rangle$ eine S5-FA-Instanz. Da der Rahmen (W, S) vollständig ist, ist jede Teilformel von φ, die mit einem modalen Operator beginnt (also Teilformeln der Form $\Box\alpha$) entweder in allen Welten aus W oder in keiner erfüllt. Auf Basis dieser Erkenntnis geben wir Algorithmus 10 an, der S5-FA entscheidet.
Von „innen nach außen“ werden Formeln der Form $\Box\alpha$ wie folgt durch Konstanten ersetzt: Gilt α in jeder Welt, ersetzen wir es durch $\top$, sonst durch $\bot$. Dies wird durch die Funktion `BOX` (ab Zeile 8) rekursiv getan. Ob eine Formel in jeder Welt gilt, muss so nur für Formeln geprüft werden, die keine modalen Operatoren enthalten – dafür wird die Funktion `FA` verwendet. Die Formel wird einmal komplett rekursiv durchlaufen, dafür gibt es maximal einen rekursiven Aufruf für jede Teilformel. Alle Variablen können in logarithmischem Platz gespeichert werden und die Laufzeit ist polynomiell. Das Auswerten einer Formel ohne modale Operatoren in einer Welt entspricht der aussagenlogischen Formelauswertung und ist in logarithmischem Platz möglich (siehe Theorem 2.31). Die Rekursion kann auf einem Stapel organisiert werden und daraus folgt, dass Algorithmus 10 auf einer Turingmaschine implementiert werden kann, der die Ressourcen von LOGdetCFL zur Verfügung stehen. □

Hierbei ist interessant, dass allein der Fakt, dass es verschiedene Welt mit beliebig vielen verschiedenen Belegungen geben kann, nicht ausreicht, um die P-Härte des

Algorithmus 10 `Formelauswertung für` S5

Eingabe: S5-Formel φ, S5-Modell $\mathcal{M}$, Welt w aus $\mathcal{M}$

1: **wenn** $\texttt{FA}(\varphi, \mathcal{M}, w)$ **dann** akzeptieren **sonst** lehne ab

2: **Funktion** $\texttt{FA}(\psi, \mathcal{N} = (V, R, \pi, \text{VAR}), v \in V)$ // bestimmt $\mathcal{N}, v \models \psi$
3: **wenn** $\psi = \top$ **dann Rückgabe** `true`
4: **wenn** $\psi = \bot$ **dann Rückgabe** `false`
5: **wenn** $\psi \in \text{VAR}$ **dann Rückgabe** $v \in \pi(\psi)$
6: **wenn** $\psi = \Box\alpha$ **dann Rückgabe** $\texttt{FA}(\texttt{BOX}(\alpha, \mathcal{N}), \mathcal{N}, v)$
7: **wenn** $\psi = \alpha \to \beta$ **dann Rückgabe** $\texttt{FA}(\alpha, \mathcal{N}, v) \Rightarrow \texttt{FA}(\beta, \mathcal{N}, v)$

8: **Funktion** $\texttt{BOX}(\psi, \mathcal{N} = (V, R, \pi, \text{VAR}))$ // bestimmt $\mathcal{N}, V \models \psi$
9: **wenn** $\psi \in \{\bot, \top\}$ **dann Rückgabe** ψ
10: **wenn** $\psi \in \text{VAR}$ **dann**
11: **wenn** $V = \pi(\psi)$ **dann Rückgabe** $\top$ **sonst Rückgabe** $\bot$
12: **wenn** $\psi = \Box\alpha$ **dann Rückgabe** $\texttt{BOX}(\alpha, \mathcal{N})$
13: **wenn** $\psi = \alpha \to \beta$ **dann**
14: **wenn** α nicht $\Box$-frei ist **dann** $\alpha := \texttt{BOX}(\alpha, \mathcal{N})$
15: **wenn** β nicht $\Box$-frei ist **dann** $\beta := \texttt{BOX}(\beta, \mathcal{N})$
16: **wenn** $\alpha \to \beta$ in allen Welten aus W gilt **dann Rückgabe** $\top$
17: **sonst Rückgabe** $\bot$

Formelauswertungsproblems zu erreichen. Die Art, wie die Welten in Verbindung stehen, ist also auch für die Komplexität ausschlaggebend, da diese Verbindungen die Mächtigkeit und die Art der Wirkung der modalen Operatoren wesentlich beeinflussen. Bei S5 sind genau genommen nur die vorkommenden Belegungen in einem Modell entscheidend. Deswegen ist auch die Formelauswertung leichter, wenn wir die Zahl der Variablen und damit auch die Zahl der möglichen Belegungen beschränken.

Theorem 6.10 *Für alle $n \in \mathbb{N}$ ist das Problem* S5[n]-FA NC^1*-vollständig.*

Beweisidee. Die untere Schranke folgt direkt aus Theorem 2.31. Es ist leicht zu sehen, dass in S5[n]-Modellen eine Welt nur höchstens 2^n viele verschiedene Belegungen haben kann. Zwei Welten in einem S5[n]-Modell mit derselben Belegung sind äquivalent, da jede Welt genau dieselben Nachbarn hat, nämlich jede Welt dieses Modells. Also kann ein Modell auch nur aus maximal 2^n vielen paarweise nicht äquivalenten Welten bestehen. Damit ist klar, dass es nur höchstens 2^{2^n} viele verschiedene S5[n]-Modelle gibt.[3] Mit Lemma 3.6 folgt, dass S5[n] endlich erzeugt ist. Wir haben in Theorem 3.3 nur für intuitionistische Logiken gezeigt, dass ihr Formelauswertungsproblem in NC^1 ist, aber der Beweis lässt sich einfach auf Modallogiken übertragen (siehe dazu Bemerkung 3.14). □

[3] Ein streng formaler Beweis dafür, dass es in S5[n] bezüglich $\equiv_{\mathfrak{F}[n]}$ nur endlich viele Äquivalenzklassen gibt, ist der Konstruktion aus dem Beweis von Theorem 3.10 (LC[n] ist endlich erzeugt) sehr ähnlich.

6.1.4 S4.3 – der modale Begleiter von LC, eine Diskussion

Für den modalen Begleiter S4.3 von LC können wir hier nur einige Vermutungen angeben. Offensichtlich ist das triviale Resultat, dass S4.3[0]-FA NC^1-vollständig ist. Es folgt aus den Theoremen 2.31 und 6.6. Für die Fragmente LC[n] mit n Variablen konnten wir zeigen, dass sie endlich erzeugt sind (Theorem 3.10). Ihr Formelauswertungsproblem ist damit ebenfalls NC^1-vollständig (folgt aus Theorem 3.3).
Lässt man hingegen in S4.3 nur eine Variable zu, so gibt es bereits unendlich viele nicht äquivalente Formeln. Wir definieren die Formeln $\varphi_0 := p$, $\varphi_1 := p \wedge \Diamond\neg p$ und $\varphi_{n+2} := p \wedge \Diamond(\neg p \wedge \Diamond\varphi_n)$ für $n \in \mathbb{N}$ und $p \in \mathrm{VAR}$. Weiter definieren wir Modelle $\mathcal{M}_n := (\{0, 1, \ldots, n\}, \leq, \xi, \{p\})$, wobei $i \in \xi(p)$ genau dann gilt, wenn $i \leq n$ und i gerade ist. Es gilt offensichtlich $\mathcal{M}_n, 0 \models \varphi_i$ für $i \leq n$ und $\mathcal{M}_n, 0 \not\models \varphi_j$ für $j > n$. Da φ_n und φ_m für $n \neq m$ nicht in derselben Äquivalenzklasse liegen, ist bereits S4.3[1] nicht endlich erzeugt. Dies legt die Vermutung nahe, dass das Formelauswertungsproblem auch nicht in NC^1 liegt.
Auch für nicht endlich erzeugte Logiken muss das Formelauswertungsproblem nicht P-hart sein. Dass IPL[1]-FA in AC^1 liegt, basiert im Wesentlichen darauf, dass die Formeln sehr groß sind im Vergleich zu den Modellen, die charakterisieren, aus welcher Äquivalenzklasse sie sind. Je größer die Formeln sind, desto einfacher ist das Formelauswertungsproblem, da die großen Formeln immer auch Bestandteil der Eingabe sind.[4] Aber auch diese Eigenschaft ist bei S4.3[1] nicht gegeben, da φ_n ähnlich groß wie $\mathcal{M}_n$ ist (beide sind polynomiell in n).
Um zu zeigen, dass S4.3[1]-FA in LOGdetCFL oder LOGCFL ist, bräuchten wir eine systematische Beschreibung der Formeläquivalenzklassen, ähnlich wie bei FPL[0]. Oder die Informationen, in welchen Welten eine Formel erfüllt ist, müssten sich kompakt kodieren lassen wie bei LC oder S5.
Insgesamt deutet vieles daraufhin, dass S4.3[1]-FA nicht in NC^1 liegt. Damit wären LC[n] und S4.3[n] für $1 \leq n \leq \infty$ ebenfalls wie S4.2[1] und IPL[1] Vertreter, bei denen das Formelauswertungsproblem im modalen Begleiter schwerer ist als in der zugehörigen intuitionistischen Logik.

6.2 Vergleich zwischen intuitionistischen Logiken und ihren modalen Begleitern

Zu welcher intuitionistischen Logik welcher modale Begleiter gehört, ist in Tabelle 6.1 dargestellt. Natürlich ist ein modaler Begleiter nicht eindeutig bestimmt, da man auch jede Logik als Begleiter nehmen könnte, die ihn enthält. Zum Beispiel könnte man auch K4 als modalen Begleiter von LC sehen. Da aber mit größeren und allgemeineren Logiken die Probleme schwerer werden, betrachten wir hier

[4] Für IPL[1] wird das auch dadurch deutlich, dass eine kompakte Repräsentation der Formeln dazu führt, dass das Formelauswertungsproblem P-hart wird (siehe dazu Abschnitt 4.5.2).

immer möglichst eingeschränkte Logiken. Deswegen wählen wir S4.3 als modalen Begleiter von LC.

Intuitionistische Logik	Modaler Begleiter	Modelleigenschaften
BPL	K4	transitiv
FPL	PrL	transitiv und irreflexiv
IPL	S4	Halbordnung
KC	S4.2	gerichtete Halbordnung
LC	S4.3	lineare Ordnung
AL	S5	totale Ordnung

Tabelle 6.1: Intuitionistische Logiken mit ihren modalen Begleitern und den entsprechenden Modelleigenschaften.

Abschließend wollen wir die Komplexität des Formelauswertungsproblems für die intuitionistischen Logiken und ihre modalen Begleiter gegenüberstellen. Wie bereits einleitend in diesem Kapitel erwähnt, gibt es in einem modalen Begleiter mehr Modelle, da diese zum Beispiel nicht kreisfrei sein müssen. Auch wird in der Modallogik nicht gefordert, dass die Belegungsfunktion monoton ist. Damit ist es nicht unmöglich, dass die Formelauswertung in modalen Begleitern schwerer als in den zugehörigen intuitionistischen Logiken ist. Im Beweis von Theorem 6.4 wurden genau diese Freiheitsgrade genutzt, um zu zeigen, dass S4.2[1]-FA P-hart ist, während KC[1]-FA in NC^1 liegt. Wir fassen jetzt die komplexitätstheoretischen Resultate für die Formelauswertung in den verschiedenen Logiken zusammen. Wenn nichts zur oberen Schranke gesagt wird, gilt P als beste obere Schranke (Theorem 2.30).

- Für BPL und K4 benötigt man keine Variablen für die P-Härte, lediglich die intuitionistische Implikation, $\vee$ und die Konstante $\bot$ sind notwendig (Theoreme 5.5 und 6.2).
- Die P-Härte für Fragmente von FPL und PrL erreicht man bereits mit intuitionistischer (bzw. strikter) Implikation, einer Variablen und $\bot$ (Theoreme 5.3 und 6.2). Verwendet man keine Variablen, so ist $\mathsf{LOGdetCFL}$ bei FPL (Theorem 5.8) und AC^1 bei PrL AC^1 (Theorem 6.7) die obere Schranke. Als untere Schranke können wir in beiden Fällen nur NL angeben (Theorem 5.9).
- Bei IPL und S4 gibt es bei den Fragmenten mit nur einer Variablen einen interessanten Unterschied. Während wir für IPL mit einer Variablen die AC^1-Vollständigkeit gezeigt haben (Theorem 4.29) und die P-Härte erst ab zwei Variablen vorliegt (Bemerkung 5.7), genügt dafür bei S4 bereits eine Variable (Bemerkung 6.5). Ohne Variablen gilt für beide Logiken NC^1 als obere Schranke (folgt aus Theorem 2.31).

- Für KC und S4.2 ist es ähnlich. Bei KC-Fragmenten ohne Variablen oder mit einer Variablen ist die Formelauswertung in NC^1 (folgt aus den Theoremen 2.31 und 4.30). Ab zwei Variablen gilt hier wieder die P-Härte (Theorem 5.6). Dagegen verhält sich S4.2 wie S4, hier erhalten wir die P-Härte bereits mit einer Variablen (Theorem 6.4).

- Für LC und S4.3 können wir nur wenige genaue Ergebnisse angeben. Die NC^1-Vollständigkeit konnten wir für alle Fragmente von LC mit beschränkter Variablenzahl zeigen (Theorem 3.13). Verzichtet man auf solch eine Beschränkung, haben wir $\mathsf{LOGdetCFL}$ als obere Schranke (Theorem 3.12), aber keine bessere untere Schranke als NC^1. Für S4.3-Fragmente ohne Variablen ist NC^1 die obere Schranke (folgt aus Theorem 2.31), aber bereits ab einer Variablen können wir nur noch Vermutungen anstellen. Unsere Diskussion (Abschnitt 6.1.4) liefert einige Hinweise, die darauf hindeuten, dass hier NC^1 nicht mehr die obere Schranke ist. Auch für S4.3 ohne Beschränkung der Variablen ist die Komplexität des Formelauswertungsproblems noch völlig offen.

- Da AL im ursprünglichen Sinne keine intuitionistischen Logik ist, haben wir AL und S5 nur am Rand betrachtet. Für AL gilt NC^1-Vollständigkeit (folgt aus Theorem 2.31) und für S5 haben wir $\mathsf{LOGdetCFL}$ als obere Schranke (Theorem 6.9). Eine bessere untere Schranke konnten wir nicht zeigen. Beschränkt man bei S5 die Zahl der Variablen, ist die Formelauswertung wieder NC^1-vollständig.

Eine interessante Feststellung bei S4[1] und S4.2[1] ist, dass bereits die strikte Implikation als Operator für die P-Härte der Formelauswertung genügt. Bisher wurden Fragmente der Modallogik meist hinsichtlich des Postschen Verbandes [Pos41] gebildet bzw. aussagenlogische und modale Operatoren getrennt betrachtet, die strikte Implikation hingegen ist eine Kombination aus beiden Typen. Auch bei PrL[0] ist die strikte Implikation für unser Härteresultat ausreichend.

Kapitel 7

Zusammenfassung

Wir haben in dieser Arbeit im Wesentlichen die Komplexität des Formelauswertungsproblems intuitionistischer Logiken untersucht. Dabei unterteilten wir die Untersuchungen in vier Kapitel. In Kapitel 3 betrachteten wir die endlichen Logiken. Kapitel 4 widmete sich hauptsächlich der intuitionistischen Logik mit einer Variablen. Logiken mit einem P-harten Formelauswertungsproblem betrachteten wir in Kapitel 5. Abschließend stellten wir die intuitionistischen Logiken ihren modalen Begleitern in Kapitel 6 gegenüber. Am Ende jedes Kapitels gibt es eine Zusammenfassung, die einen Überblick über die Resultate des Kapitels enthält.
Auf der einen Seite konnten wir für viele Fragmente zeigen, dass ihr Formelauswertungsproblem effizient parallelisierbar – also in NC^1 – ist. Auf der anderen Seite gibt es eine Reihe Fragmente, bei denen die Formelauswertung P-hart ist. Für weitere Fragmente liegt die Komplexität dazwischen. Neben der Einordnung in Komplexitätsklassen hat uns auch die Abgrenzung interessiert. Dabei stand die Frage im Mittelpunkt, wie viele Variablen man für die P-Härte benötigt. Dazu liefert Tabelle 7.1 eine Übersicht.

Logik / Variablenzahl	AL[n]	LC[n]	KC[n]	IPL[n]	FPL[n]	BPL[n]
$n = 0$	NC^1	NC^1	NC^1	NC^1	NL-hart $\in$ LOGCFL	P
$n = 1$	NC^1	NC^1	NC^1	AC^1	P	P
$n \geq 2$	NC^1	NC^1	P	P	P	P
$n = \infty$	NC^1	NC^1-hart $\in$ LOGdetCFL	P	P	P	P

Tabelle 7.1: Die Komplexität des Formelauswertungsproblems hängt bei den intuitionistischen Logiken neben der Wahl der Logik auch stark von der Anzahl der gewählten Variablen ab.

Besonders hervorzuheben ist das Resultat aus Kapitel 4. Für IPL[1]-FA konnten wir die AC^1-Vollständigkeit zeigen. Für den Beweis der AC^1-Härte verwendeten wir genauso wie in den Beweisen für die P-Härte aus Kapitel 5 alternierende Pfade durch Schichtgraphen als Basis. Dieser Zusammenhang ist darin begründet,

dass sich sowohl AC^1, als auch P als eine alternierende Logspaceklasse charakterisieren lassen – $\mathsf{P} = \mathsf{ALOGSPACE}[n^{\mathcal{O}(1)}]$ und $\mathsf{AC}^1 = \mathsf{ALOGSPACE}[\log(n)]$. Die AC^1-Vollständigkeit von IPL[1]-FA ist insofern interessant, als dass es das erste Problem ist, bei dem die logarithmische Beschränkung der Zahl der Alternierungen kein Bestandteil der Problemdefinition ist.
Auch Kapitel 3 lieferte ein sehr interessantes Resultat. Für NC^1 als obere Schranke des Formelauswertungsproblems ist es hinreichend zu zeigen, dass die Logik endlich erzeugt ist. Damit bekommt die Heyting-Semantik eine neue Bedeutung in Bezug auf die Komplexität der Formelauswertung in der Kripke-Semantik, denn sie basiert auf den Formeläquivalenzklassen einer Logik. Ist die Heyting-Algebra einer Logik endlich, so ist diese Logik endlich erzeugt und ihr Formelauswertungsproblem effizient parallelisierbar. Diese Komplexität überträgt sich auch auf das Erfüllbarkeits- und das Tautologieproblem dieser Logik.
Für einige Fragmente konnten wir kein Vollständigkeitsresultat angeben. Zu nennen sind hier vor allem die Logik LC und ihr modaler Begleiter S4.3. Obwohl die scheinbar sehr einfache Struktur der Modelle daraufhin deutet, dass die Formelauswertung einfach ist, ist die Komplexität für S4.3[n]-FA für $n \in \mathbb{N}^+ \cup \{\infty\}$ noch völlig offen. Ein weiteres interessantes Feld ist die Kombination von aussagenlogischen und modalen Operatoren. Für einige Logiken haben wir gezeigt, dass die strikte Implikation bereits für die P-Härte der Formelauswertung ausreichend ist. Bisher wurden Fragmente der Modallogik meist hinsichtlich des Postschen Verbandes gebildet. Völlig offen ist dagegen die Frage, ob es weitere Fragmente mit kombinierten Operatoren gibt, für die man interessante Komplexitätsresultate erhält.

Literaturverzeichnis

[Bal73] R. Balbes. On free pseudo-complemented and relatively pseudo-complemented semi-lattices. *Fundamenta Mathematicae*, 78:119–131, 1973.

[BCGR90] S. R. Buss, S. A. Cook, A. Gupta und V. Ramachandran. An optimal parallel algorithm for formula evaluation. *SIAM Journal on Computing*, 21:755–780, 1990.

[BdV01] P. Blackburn, M. de Rijke und Y. Venema. *Modal Logic*, Band 53 in *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2001.

[Bet47] E. W. Beth. Semantic considerations on intuitionistic mathematics. *Indagationes Mathematicae*, 9:572–577, 1947.

[Bet56] E. W. Beth. Semantic construction of intuitionistic logic. *Koninklijke Nederlandse Akademie van Wetenschappen, Mededelingen, Nieuwe Reeks*, 19(11):357–388, 1956.

[Blu92] A. L. Blum. *Algorithms for approximate graph coloring*. Dissertation, Cambridge, 1992.

[BM95] M. Beaudry und P. McKenzie. Circuits, matrices, and nonassociative computation. *Journal of Computer and System Sciences*, 50(3):441–455, 1995.

[BMS+11] M. Bauland, M. Mundhenk, T. Schneider, H. Schnoor, I. Schnoor und H. Vollmer. The tractability of model checking for LTL: the good, the bad, and the ugly fragments. *ACM Transactions on Computational Logic (TOCL)*, 12(2), 2011.

[Boo93] G. S. Boolos. *The Logic of Provability*. Cambridge University Press, 1993.

[Bou04] F. Bou. Complexity of strict implication. In *Advances in Modal Logic 5*, Seiten 1–16, 2004.

[Bro18] L.E.J. Brouwer. *Begründung der Mengenlehre unabhängig vom logischen Satz vom ausgeschlossenen Dritten. Erster Teil: Allgemeine*

Mengenlehre. Verhandelingen der Koninklijke akademie van wetenschappen te Amsterdam, (1. sectie) Deel XII, no. 5, 7, Amsterdam, 1918.

[Bro25] L.E.J. Brouwer. Intuitionistische Zerlegung mathematischer Grundbegriffe. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 33:251–256, 1925.

[Bus87] S. R. Buss. The Boolean formula value problem is in ALOGTIME. In *19th ACM STOC*, Seiten 123–131, ACM, 1987.

[BvBW06] P. Blackburn, J. van Benthem und F. Wolter. *Handbook of Modal Logic*. North-Holland, Amsterdam, 2006.

[Cha85] A. V. Chagrov. On the complexity of propositional logics. In *Complexity Problems in Mathematical Logic*, Seiten 80–90, Kalinin State University, 1985. Auf Russisch.

[Chu36] A. Church. A note on the Entscheidungsproblem. *Journal of Symbolic Logic*, 1:101–102, 1936.

[CKS81] A. K. Chandra, D. Kozen und L. J. Stockmeyer. Alternation. *Journal of the Association for Computing Machinery*, 28:114–133, 1981.

[CLR90] T. H. Cormen, C. E. Leiserson und R. L. Rivest. *Introduction to Algorithms*. The MIT Electrical Engineering and Computer Science Series. MIT Press, Cambridge, MA, 1990.

[Coo71a] S. A. Cook. Characterizations of pushdown machines in terms of time-bounded computers. *ACM*, 18:4–18, 1971.

[Coo71b] S. A. Cook. The complexity of theorem-proving procedures. In *3rd ACM STOC*, Seiten 151–158, ACM, 1971.

[Coo85] S. A. Cook. A taxonomy of problems with fast parallel algorithms. *Information and Control*, 64:2–22, 1985.

[CR02] A. V. Chagrov und M. N. Rybakov. How many variables does one need to prove pspace-hardness of modal logics. In *Advances in Modal Logic 4*, Seiten 71–82, 2002.

[CZ97] A. V. Chagrov und M. Zakharyaschev. *Modal Logic*. Clarendon Press, Oxford, 1997.

[Die06] R. Diestel. *Graphentheorie*. Springer, 3. Auflage, Berlin, Heidelberg, 2006.

[DL59] M. Dummett und E. Lemmon. Modal logics between S4 and S5. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 14(24):250–264, 1959.

[dLHdJ12] G. R. R. de Lavalette, A. Hendriks und D. H. J. de Jongh. Intuitionistic implication without disjunction. *Journal of Logic and Computation*, 22(3):375–404, 2012.

[Dum59] M. Dummett. A propositional calculus with denumerable matrix. *Journal of Symbolic Logic*, 24(2):97–106, 1959.

[End01] H. B. Enderton. *A mathematical introduction to logic*. Hartcourt/Academic Press, 2. Auflage, 2001.

[Fag74] R. Fagin. Generalized first-order spectra and polynomial-time recognizable sets. *Complexity of Computation*, 7:43–74, 1974.

[FdR06] M. Franceschet und M. de Rijke. Model checking for hybrid logics (with an application to semistructured data). *Journal of Applied Logic*, 4(3):279–304, 2006.

[FL79] M. J. Fischer und R. E. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and Systems Sciences*, 18(2):194–211, 1979.

[Gab81] D. M. Gabbay. *Semantical investigations in Heyting's intuitionistic logic*. D. Reidel Publishing Company, Dordrecht, Boston, London, 1981.

[Göd32] K. Gödel. Zum intuitionistischen Aussagenkalkül. *Anzeiger der Akademie der Wissenschaften in Wien, Mathematisch- Naturwissenschaftliche Klasse*, 69:65–66, 1932.

[Gen34] G. Gentzen. Untersuchungen über das logische schließen. *Mathematische Zeitschrift*, 39:176–210, 405–431, 1934.

[GHR95] R. Greenlaw, H. J. Hoover und W. L. Ruzzo. *Limits to Parallel Computation: P-Completeness Theory*. Oxford University Press, New York, 1995.

[Gol06] R. Goldblatt. Mathematical modal logic: A view of its evolution. In *Handbook of the History of Logic 7*, herausgegeben von D. M. Gabbay und J. Woods, Seiten 1–98, Elsevier, 2006.

[Hey30] A. Heyting. Die formalen Regeln der intuitionistischen Logik. *Sitzungsberichte der preußischen Akademie der Wissenschaften, phys.-math. Klasse*, Seiten 42–65, 1930.

[Hey31] A. Heyting. Die intuitionistische Grundlegung der Mathematik. *Erkenntnis*, 2:106–115, 1931.

[Hey34] A. Heyting. *Mathematische Grundlagenforschung. Intuitionismus, Beweistheorie*. Ergebnisse der Mathematik und ihre Grenzgebiete. Springer, Berlin, 1934.

[Hey71] A. Heyting. *Intuitionism*. North-Holland, Amsterdam, 1971.

[Hey86] A. Heyting. Die formalen Regeln der intuitionistischen Logik (gekürzter Nachdruck). In *Logik-Texte: Kommentierte Auswahl zur Geschichte der Modernen Logik*, herausgegeben von K. Berka und L. Kreiser Seiten 188–192, Akademie-Verlag, 4. Auflage, Berlin, 1986.

[HM92] J. Halpern und Y. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54(2):319–379, 1992.

[HSS10] E. Hemaspaandra, H. Schnoor und I. Schnoor. Generalized modal satisfiability. *Journal of Computer and System Sciences*, 76(7):561–578, 2010.

[Imm79] N. Immerman. Length of predicate calculus formulas as a new complexity measure. In *20th FOCS*, Seiten 337–347, IEEE Computer Society, 1979.

[Imm82] N. Immerman. Upper and lower bounds for first order expressibility. *Journal of Computer and System Sciences*, 25:76–98, 1982.

[Imm86] N. Immerman. Relational queries computable in polynomial time. *Information and Control*, 68:86–104, 1986.

[Imm89] N. Immerman. Descriptive and computational complexity. In *Computational Complexity Theory, Symposium on Applied Mathematics*, Seiten 75–91, American Mathematical Society, 1989.

[Imm99] N. Immerman. *Descriptive Complexity*. Graduate Texts in Computer Science. Springer, New York, Berlin, Heidelberg, 1999.

[Ind10] A. Indrzejczak. *Natural Deduction, Hybrid Systems and Modal Logics*. Springer, Dordrecht, Heidelberg, London, New York, 2010.

[Jas36] S. Jaskowski. Recherches sur le système de logique intuitioniste. In *Actes du Congrès International de Philosophie Scientifique (Sorbonne, Paris, 1935), VI, Philosophie des Mathématiques*, Seiten 58–61, 1936.

[Joh82] P. T. Johnstone. *Stone spaces*. Cambridge University Press, Cambridge, 1982.

[JT07] A. Jakoby und T. Tantau. Logspace algorithms for computing shortest and longest paths in series-parallel graphs. In *FSTTCS*, Band 4855 in *LNCS*, Seiten 216–227, 2007.

[Kol32] A. N. Kolmogorov. Zur Deutung der intuitionistischen Logik. *Mathematische Zeitschrift*, 35:58–65, 1932.

[KR90] R. Karp und V. Ramachandran. Parallel algorithms for shared-memory machines. In *Handbook of Theoretical Computer Science A*, herausgegeben von J. van Leeuwen, Seiten 869–941, Elsevier, 1990.

[Kri63a] S. Kripke. A semantical analysis of modal logic I: Normal modal propositional calculi. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 9:67–96, 1963.

[Kri63b] S. Kripke. Semantical considerations on modal and intuitionistic logic. *Acta Philosophica Fennica*, 16:83–94, 1963.

[Kri65] S. Kripke. Semantical analysis of intuitionistic logic I. In *8th Logics Colloquium*, Seiten 92–130, 1965.

[Lad75] R. Ladner. The circuit value problem is log space complete for P. *SIGACT News*, 7(1):12–20, 1975.

[Lad77] R. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM Journal on Computing*, 6(3):467–480, 1977.

[Lew18] C. I. Lewis. *A survey of symbolic logic*. University of California Press, Berkley, 1918.

[Lyn77] N. Lynch. Log space recognition and translation of parenthesis languages. *Journal of the Association for Computing Machinery*, 24(4):583–590, 1977.

[Mar04] N. Markey. Past is for free: on the complexity of verifying linear temporal properties with past. *Acta Informatica*, 40(6-7):431–458, 2004.

[McK68] G. McKay. The decidability of certain intermediate propositional logics. *Journal of Symbolic Logic*, 33:258–264, 1968.

[Mei11] A. Meier. *On the Complexity of Modal Logic Variants and their Fragments*. Dissertation, Leibniz Universität Hannover, 2011.

[MMTV09] A. Meier, M. Mundhenk, M. Thomas und H. Vollmer. The complexity of satisfiability for fragments of CTL and CTL*. *International Journal of Foundations of Computer Science*, 20(5):901-918, 2009.

[MT44] J. C. C. McKinsey und A. Tarski. The algebra of topology. *Annals of Mathematics*, 45:141–191, 1944.

[MT46] J. C. C. McKinsey und A. Tarski. On closed elements in closure algebra. *Annals of Mathematics*, 47:122–162, 1946.

[MW10] M. Mundhenk und F. Weiß. The complexity of model checking for intuitionistic logics and their modal companions. In *4th Workshop on Reachability Problems*, Band 6227 in *LNCS*, Seiten 146–160, Springer, 2010.

[MW11] M. Mundhenk und F. Weiß. The model checking problem for intuitionistic propositional logic with one variable is AC^1-complete. In *28th STACS*, Band 9 in *LIPIcs*, Seiten 368–379, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2011.

[MW12a] M. Mundhenk und F. Weiß. An AC^1-complete model checking problem for intuitionistic logic. 2012, erscheint in Kürze in *Computational Complexity*.

[MW12b] M. Mundhenk und F. Weiß. Intuitionistic implication makes model checking hard. *Logical Methods in Computer Science*, 8(2), 2012.

[Nis60] I. Nishimura. On formulas of one variable in intuitionistic propositional calculus. *Journal of Symbolic Logic*, 25:327–331, 1960.

[NS97] A. Nerode und R. A. Shore. *Logic for Applications*. Springer, 2. Auflage, New York, Berlin, Heidelberg, London, 1997.

[Pap94] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, Reading, MA, 1994.

[Pos41] E. Post. The two-valued iterative systems of mathematical logic. *Annals of Mathematical Studies*, 5:1–122, 1941.

[Rei11] K. Reinhardt. Model checking for PrL_0 on linear frames is in AC^1. Unveröffentlicht, 2011.

[Ruz80] W. L. Ruzzo. Tree-size bounded alternation. *Journal of Computer and System Sciences*, 21(2):218–235, 1980.

[Ruz81] W. L. Ruzzo. On uniform circuit complexity. *Journal of Computer and Systems Sciences*, 21:365–383, 1981.

[Ryb06] M. N. Rybakov. Complexity of intuitionistic and Visser's basic and formal logics in finitely many variables. In *Advances in Modal Logic 6*, Seiten 393–411, College Publications, 2006.

[SC85] A. Sistla und E. Clarke. The complexity of propositional linear temporal logics. *Journal of the ACM*, 32(3):733–749, 1985.

[Sch00] U. Schöning. *Logik für Informatiker*. Spektrum Akademischer Verlag, 5. Auflage, 2000.

[Sch08] U. Schöning. *Theoretische Informatik - kurz gefasst*. Spektrum Akademischer Verlag, 5. Auflage, 2008.

[Sch02] T. Schneider. *Komplexität modaler Logiken*. Diplomarbeit, Friedrich-Schiller-Universität Jena, 2002.

[Sch07a] T. Schneider. *The Complexity of Hybrid Logics over Restricted Classes of Frames*. Dissertation, Friedrich-Schiller-Universität Jena, 2007.

[Sch07b] H. Schnoor. *Algebraic Techniques for Satisfiability Problems*. Dissertation, Gottfried Wilhelm Leibniz Universität Hannover, 2007.

[Sch10] H. Schnoor. The complexity of model checking for boolean formulas. *International Journal of Foundations of Computer Science*, Seiten 289–309, 2010.

[Sko53] T. A. Skolem. Considerations on the foundations of mathematics. *Revista matemática hispano-americana*, 12:169–200, 1952; 13:149–174, 1953. Auf Spanisch.

[Smo87] R. Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *19th ACM STOC*, Seiten 77–82, ACM, 1987.

[Spa93] E. Spaan. *Complexity of Modal Logics*. Dissertation, University of Amsterdam, 1993.

[SS89] G. Schmidt und T. Strohlein. *Relationen und Graphen*. Springer, Berlin, 1989.

[Sta79] R. Statman. Intuitionistic propositional logic is polynomial-space complete. *Theoretical Computer Science*, 9:67–72, 1979.

[Sto37] M. H. Stone. Topological representations of distributive lattices and Brouwerian logics. *Casopis Pro Pěstovánˊi Matematiky a Fysiki*, 67:1–25, 1937.

[Sto74] L. J. Stockmeyer. *The complexity of decision problems in automata theory and logic*. Dissertation, Massachusetts Institute of Technology, 1974.

[Sve03a] V. Švejdar. The decision problem of provability logic with only one atom. *Archive for Mathematical Logic*, 42(8):763–768, 2003.

[Sve03b] V. Švejdar. On the polynomial-space completeness of intuitionistic propositional logic. *Archive for Mathematical Logic*, 42(7):711–716, 2003.

[Tar38] A. Tarski. Der Aussagenkalkül und die Topologie. *Fundamenta Mathematicae*, 31:103–134, 1938.

[tCF05] B. ten Cate und M. Franceschet. On the complexity of hybrid logics with binders. In *Proceedings of Computer Science Logic 2005*, Band 3634 in *LNCS*, Seiten 339–354, Springer, 2005.

[Tur37] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. In *Proceedings of the London Mathematical Society*, Seiten 42:230–265, Korrekturen 43:544–546, 1937.

[Urq74] A. Urquhart. Implicational formulas in intuitionistic logic. *Journal of Symbolic Logic*, 39(4):661–664, 1974.

[vD04] D. van Dalen. *Logic and Structure*. Springer, 4. Auflage, Berlin, Heidelberg, 2004.

[vDT88] D. van Dalen und A. S. Troelstra. *Constructivism in Mathematics, Vol. I & II*. North-Holland, Amsterdam, 1988.

[Vis80] A. Visser. A propositional logic with explicit fixed points. *Studia Logica*, 40:155–175, 1980.

[Vol99] H. Vollmer. *Introduction to Circuit Complexity – A Uniform Approach*. Texts in Theoretical Computer Science. Springer, Berlin Heidelberg, 1999.

[Wec00] G. Wechsung. *Vorlesung zur Komplexitätstheorie*. Teubner-Texte zur Informatik. B. G. Teubner, Stuttgart, Leipzig, Wiesbaden, 2000.

[Wei08] F. Weiß. *Ausgewählte Komplexitätsprobleme der Modallogik und der intuitionistischen Aussagenlogik*. Diplomarbeit, Friedrich-Schiller-Universität Jena, 2008.

Ehrenwörtliche Erklärung

Hiermit bestätige ich, dass

- mir die geltende Promotionsordnung bekannt ist,
- ich die Dissertation selbst angefertigt habe, keine Textabschnitte oder Ergebnisse eines Dritten oder eigener Prüfungsarbeiten ohne Kennzeichnung übernommen und alle von mir benutzten Hilfsmittel, persönlichen Mitteilungen und Quellen in der vorgelegten Dissertation angegeben habe,
- ich nicht die Hilfe eines Promotionsberaters in Anspruch genommen habe und dass Dritte weder unmittelbar noch mittelbar geldwerte Leistungen von mir für Arbeiten erhalten haben, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen,
- ich die Dissertation noch nicht als Prüfungsarbeit für eine staatliche oder andere wissenschaftliche Prüfung eingereicht habe und
- ich weder Teile noch die gesamte Arbeit bei einer anderen Hochschule als Dissertation eingereicht habe.

Jena, den 08.01.2013

Felix Weiß