

DFG-Antrag auf Gewährung einer Sachbeihilfe für das Projekt

Generische algorithmische Methoden in der Modallogik

Lutz Schröder
DFKI-Labor Bremen

28. August 2007

Inhaltsverzeichnis

1	Allgemeine Angaben	1
1.1	Antragsteller	1
1.2	Thema	1
1.3	Fachgebiet und Arbeitsrichtung	1
1.4	Voraussichtliche Gesamtdauer	1
1.5	Antragszeitraum	1
1.6	Zusammenfassung	2
2	Stand der Forschung, eigene Vorarbeiten	2
2.1	Stand der Forschung	2
2.1.1	Modallogik in der Informatik	2
2.1.2	Algorithmische Verfahren für Modallogik	3
2.1.3	Modallogische Beweiser	4
2.1.4	Koalgebra	4
2.1.5	Koalgebraische Modallogik	5
2.2	Eigene Vorarbeiten	5
2.2.1	Koalgebraische Modellierung	5
2.2.2	Koalgebraische Modallogik	5
3	Ziele und Arbeitsprogramm	6
3.1	Ziele	6
3.2	Arbeitsprogramm	9
3.2.1	AP1: Infrastruktur für ein generisches Beweiswerkzeug	9
3.2.2	AP2: Endliche Modelle für kovarietale Modallogiken	9
3.2.3	AP3: Generische Tableau-Algorithmen für kovarietale Modallogiken	10
3.2.4	AP4: Generische semantikbasierte PSPACE-Algorithmen	10
3.2.5	AP5: Instanzen der generischen PSPACE-Algorithmen	11
3.2.6	AP6: Pseudomodelle und EXPTIME-Algorithmen für koalgebraisches CTL	11
3.2.7	AP7: Instanzen für koalgebraisches CTL	12
3.2.8	AP8: Optimierungsvorstudie	12
4	Beantragte Mittel	13
5	Voraussetzungen für die Durchführung des Vorhabens	13
6	Erklärungen	14
7	Unterschrift	14
8	Verzeichnis der Anlagen	15

1 Allgemeine Angaben

Antrag auf Gewährung einer Sachbeihilfe, Neuantrag

1.1 Antragsteller

PD Dr. Lutz Schröder

Senior Researcher

(befristetes Arbeitsverhältnis bis 31.12.2008, danach abhängig von Evaluation des DFKI-Labors Bremen Verlängerung oder Rückkehr auf ruhende Assistentenstelle an der Universität Bremen)

geb. 15.08.1970, Deutscher

Früherer Antrag auf Förderung unter Geschäftszeichen SCHR 1118/2-1

DFKI-Labor Bremen

Forschungsgruppe Sichere Kognitive Systeme

Dienstliche Adresse

DFKI-Labor Bremen

Robert-Hooke-Str. 5, 28359 Bremen

Tel. (0421)218-64216, Fax: (0421)218-9864216

E-Mail Lutz.Schroeder@dfki.de

Privatadresse

Emmastr. 273

28213 Bremen

Tel. (0421)354106

1.2 Thema

Generische algorithmische Verfahren und Komplexitätsschranken für Modallogik auf koalgebraischer Basis

1.3 Fachgebiet und Arbeitsrichtung

Fachgebiet: Theoretische Informatik

Arbeitsrichtung: Entscheidungsverfahren in der Modallogik

1.4 Voraussichtliche Gesamtdauer

3 Jahre

1.5 Antragszeitraum

3 Jahre; gewünschter Beginn der Förderung 1.1.2008.

1.6 Zusammenfassung

Modallogiken verschiedenster Art spielen in der Informatik und der künstlichen Intelligenz eine zentrale Rolle, etwa als Beschreibungsmittel für reaktive Systeme oder als Wissensrepräsentationslogiken. Das Interesse an Modallogiken ist insbesondere begründet in ihrer (für Verhältnisse der Logik) zum Teil relativ niedrigen Komplexität, die effektives automatisches Schließen erlaubt. Der Ad-hoc-Entwurf von Entscheidungsverfahren für gegebene Modallogiken ist jedoch häufig mit erheblichem Aufwand verbunden; dies betrifft insbesondere die in zunehmenden Maße auftretenden nicht-normalen¹ Logiken, z.B. für Wissensrepräsentation und nichtmonotones Schließen, deren Semantik auf von den klassischen Kripke-Modellen zum Teil stark abweichenden Systemtypen basiert.

Abhilfe schafft hier das allgemeine semantische Rahmenwerk der *koalgebraischen Modallogik*, das insbesondere generische algorithmische Verfahren bereitstellt, die vergleichsweise leicht für konkrete Modallogiken zu instantiieren sind. Der bisherige Entwicklungsstand dieser Theorie erlaubt im wesentlichen die Behandlung von Logiken, deren Axiomatisierung nur über den jeweils nächsten Transitionsschritt Aussagen macht. Ziel des hier beantragten Projekts ist die Ausdehnung des Anwendungsbereichs der generischen Theorie, zum einen auf allgemeinere (d.h. mehrere Schritte in die Zukunft blickende) Axiomenformate (*frame conditions*) und zum anderen auf durch Fixpunktoperatoren gekennzeichnete generische Temporallogiken. Eine experimentelle Implementierung der so gewonnenen Algorithmen liefert einen prototypischen generischen modalen Beweiser, an dem Vorstudien zu heuristischen Optimierungen durchgeführt werden. Diese Resultate bilden die Grundlage für die Entwicklung effizienter modaler Beweiswerkzeuge, die sich durch *breite Anwendbarkeit*, *Erweiterbarkeit* und *Modularität* sowie damit letztlich auch durch bessere Wartbarkeit und erhöhte Qualität auszeichnen.

2 Stand der Forschung, eigene Vorarbeiten

2.1 Stand der Forschung

2.1.1 Modallogik in der Informatik

Modallogiken zeichnen sich aus durch über die üblichen aussagenlogischen Junktoren hinausgehende logische Operatoren, die eine qualifizierte Gültigkeit bestimmter Aussagen zusichern (oder allgemeiner ein qualifiziertes Verhältnis zwischen mehreren Aussagen), etwa Gültigkeit „notwendigerweise“, „bekanntermaßen“, „für immer“ etc. In vielen Gebieten der Informatik und der künstlichen Intelligenz spielen Modallogiken eine stetig wachsende Rolle, unter anderem aufgrund ihrer Fähigkeit, konzise Ausdrucksmittel für komplexe, aber gut umrissene Problemgebiete zur Verfügung zu stellen. Der Anwendungszweck einer Logik bestimmt hierbei die Lesart und formale Interpretation der modalen Operatoren; genannt seien hier exemplarisch folgende Logiken, die notwendig nur einen Ausschnitt des Anwendungsspektrums darstellen:

¹Eine Modallogik heißt *normal*, wenn sie das *K*-Axiom $\Box(a \rightarrow b) \rightarrow (\Box a \rightarrow \Box b)$ beinhaltet.

Temporallogik Modaloperatoren der Lesart „ab jetzt immer“, „letztendlich“ u.ä. finden Anwendung in der Spezifikation und Verifikation reaktiver Systeme und nebenläufiger Programme, aufbauend u.a. auf grundlegenden Arbeiten von Manna und Pnueli [53, 40] bzw. Clarke und Emerson [8]. Eine Variante ist Pratt's dynamische Logik [54], die Modaloperatoren explizit über Programme indiziert.

Epistemische Logik Operatoren der Lesart „Agent a weiß/glaubt, dass ...“ werden etwa in der Protokollverifikation [21] sowie in der künstlichen Intelligenz für Zwecke der Repräsentation von (sicherem oder unsicherem) Wissen verwendet [13, 19]; im Falle des unsicheren Wissens gibt es hierbei eine große Variationsbreite hinsichtlich der Semantik, basierend auf der Wahl eines Typs von Unsicherheitsmaßen.

Deontische Logik Ursprünglich für Zwecke der praktischen Ethik und der formalen Analyse von rechtlichen Verpflichtungen entwickelt [74], spielen deontische Modaloperatoren der Lesart „es sollte (moralisch, rechtlich, ...) der Fall sein, dass“ bzw. „es ist die Verpflichtung von Agent a , dass“ mittlerweile eine Rolle in der Theorie der Multiagentensysteme (s. z.B. [39]) und der Koordination von Web Services [70].

Weitere informatisch relevante Modallogiken sind z.B. gradierte Modallogik [14] mit Anwendungen im Bereich der Wissensrepräsentation und der Deskriptionslogiken [71, 46], probabilistische Modallogik [25, 38, 13], und Konditionallogik, letztere insbesondere als Basislogik für nichtmonotones Schließen (s. etwa [75]).

2.1.2 Algorithmische Verfahren für Modallogik

Durch die meist auf spezielle Anwendungsszenarien ausgerichtete Beschränkung der Ausdrucksstärke sind Modallogiken häufig algorithmischer Behandlung relativ gut zugänglich. Zu den zentralen algorithmischen Problemen in diesem Bereich zählen zum einen das Modellprüfungsproblem, d.h. zu entscheiden, ob ein vorgegebenes Modell eine bestimmte Formel erfüllt, und zum anderen das Erfüllbarkeitsproblem, d.h. zu entscheiden ob eine gegebene Formel ein Modell besitzt. Im hier beantragten Projekt liegt das Hauptaugenmerk auf dem Erfüllbarkeitsproblem bzw. dem hierzu dualen Gültigkeitsproblem sowie deren beweistheoretischen (bei vollständiger Axiomatisierung äquivalenten) Entsprechungen, Konsistenz bzw. Beweisbarkeit; diese Probleme sind insbesondere zentral für den Umgang mit modalen Wissensbasen, indem sie z.B. das Subsumptionsproblem umfassen [27].

Für zahlreiche Modallogiken ist das Erfüllbarkeitsproblem durch dedizierte Algorithmen entscheidbar, häufig in vergleichsweise niedrigen Komplexitätsklassen (berücksichtigend, dass bereits das Erfüllbarkeitsproblem der Aussagenlogik NP-vollständig ist, was aber der Entwicklung effektiver SAT-Werkzeuge, die durch heuristische Optimierungsstrategien in typischen Anwendungsfällen gute Antwortzeiten erzielen, nicht entgegensteht, s. als Überblick etwa [31]). Die Komplexität (des Erfüllbarkeitsproblems) von Logiken ohne Fixpunktoperatoren ist oft nur PSPACE. Die grundlegende Arbeit in dieser Hinsicht ist der Beweis der PSPACE-Vollständigkeit klassischer Modallogiken wie K , KT und $S4$ durch Ladner [37]; eine vertiefte Darstellung, erweitert durch zusätzliche Ergebnisse für den Fall mit mehreren Agenten, findet sich in [20]. Ein Beweis

der PSPACE-Vollständigkeit probabilistischer Modallogiken wird skizziert in [13]. Die PSPACE-Vollständigkeit verschiedener epistemischer Logiken wird in [72] gezeigt. Gerade in jüngster Zeit findet sich vermehrte Forschungstätigkeit in dieser Richtung; dies schließt z.B. Beweise der PSPACE-Vollständigkeit der gradierten Modallogik [67] (deren Komplexität zuvor eher als EXPTIME vermutet worden war), verschiedener Konditionallogiken [47], der Koalitionslogik [51] und der Presburger-Modallogik [10] ein.

Die Komplexität von Modallogiken mit Fixpunktoperatoren, wie etwa der propositionalen dynamischen Logik (PDL) [54], der Computation Tree Logic (CTL) [8], oder des (die vorangehenden umfassenden) modalen μ -Kalküls [30], ist häufig EXPTIME [15, 11, 12]. Dies lässt sich den ersten beiden Fällen mit Hilfe elementarer Modellkonstruktionen zeigen, während die EXPTIME-Entscheidbarkeit des μ -Kalküls mit automatentheoretischen Mitteln bewiesen wird. Letzteres Resultat lässt sich auf Erweiterungen des μ -Kalküls insbesondere um gradierte Modaloperatoren ausdehnen [32, 5].

2.1.3 Modallogische Beweiser

Motiviert durch die Entwicklung des Semantic Web sind in jüngerer Zeit zahlreiche effektive vollautomatische Beweiser für verschiedene Deskriptionslogiken, d.h. bestimmte multimodale Logiken mit Individuen, entwickelt worden. Während sich automatische modallogische Beweise in einigen Fällen durch Zwischenschalten geeigneter Übersetzungen in Beweisern für Logik erster Stufe führen lassen [9], sind für das hier beantragte Projekt vor allem solche Beweiser von Interesse, die dedizierte modale Tableaurekalküle implementieren, wie etwa FaCT++ [69], ModLeanK [2], Pellet [65], oder RACER [18]. Die Effektivität der Beweisprozeduren wird durch Optimierungen der Tableaustrategie erreicht, s. etwa [27, 26, 68, 65].

2.1.4 Koalgebra

Die Koalgebra hat sich in den letzten Jahren zunehmend als Beschreibungsmittel der Wahl für reaktive Prozesse auf einem angemessenen Allgemeinsniveau etabliert. Hierbei ist der Begriff der Koalgebra dual zum im Bereich der funktionalen Korrektheit verwendeten Begriff der Algebra, in dem Sinne, dass destruirende und nicht konstruierende Operationen als primitiv angesehen werden, unter dem Blickwinkel von Beobachtungsoperationen auf Zuständen. Die Koalgebra bietet u.a. eine allgemeine Perspektive auf Begriffe wie Bisimulation, Koinduktion und Korekursion [56]. Der Begriff der Koalgebra subsumiert neben klassischen Transitionssystemen auch zahlreiche komplexere Systemtypen; z.B. können Zustandsübergänge auf verschiedene Weise gewichtet sein oder als durch einzelne Agenten oder Koalitionen bewirkt aufgefasst werden. Wichtig ist hierbei unter anderem die Tatsache, dass Systemtypen ähnlich wie Datentypen modular aufgebaut werden können; z.B. findet sich in [1] eine Übersicht über aus zwei Grundbausteinen aufgebaute Systemtypen, die nicht-deterministische und probabilistische Verzweigung kombinieren, wie etwa reaktive und generative probabilistische Automaten oder Segala-Systeme.

2.1.5 Koalgebraische Modallogik

Die erste allgemeine Formulierung einer generischen Modallogik für koalgebraisch modellierte reaktive Systeme ist die *koalgebraische Logik* [42], die allerdings syntaktisch wie semantisch von üblichen Lesarten stark abweicht und insofern für Anwendungszwecke eher ungeeignet ist. Lesbare Modallogiken für sehr spezielle Systemtypen finden sich in [29, 55, 36]. Die allgemeingültige Formulierung einer koalgebraischen Interpretation von Modallogik in üblicher Syntax ist die *koalgebraische Modallogik* [50]. Dieser Begriff subsumiert zahlreiche gängige Modallogiken, insbesondere viele der (und bei naheliegender Verallgemeinerung alle) in Abschnitt 2.1.1 genannten Logiken. Zahlreiche Resultate der klassischen Modallogik lassen sich auf die Allgemeinheitsebene der koalgebraischen Modallogik anheben, wie etwa die Hennessy-Milner-Eigenschaft [50], schwache Vollständigkeit [49], Dualitätstheorie [33] und Ultrafiltererweiterungen [34]. Koalgebraische Modallogiken lassen sich ferner längs des modularen Aufbaus von Systemtypen (s. Abschnitt 2.1.4) modular kombinieren [7].

Ein noch sehr eingeschränktes Entscheidbarkeitsresultat für koalgebraische Modallogiken findet sich in [49]. Für eine Fixpunktversion der koalgebraischen Logik (nicht aber der koalgebraischen Modallogik, s.o.) sind Teile des automatentheoretischen Ansatzes (s. Abschnitt 2.1.2) verallgemeinert worden [35, 73], ohne allerdings hieraus obere Komplexitätsschranken zu gewinnen.

2.2 Eigene Vorarbeiten

2.2.1 Koalgebraische Modellierung

Der Antragsteller hat mitgewirkt am Entwurf der algebraisch-koalgebraischen Spezifikationssprache COCASL, die die algebraische Standardsprache CASL [3, 45] um koalgebraische Konstrukte für die Definition von Prozesstypen und modallogische Ausdrucksmittel erweitert [44, 60]. Teil der Werkzeugunterstützung von COCASL ist eine gut automatisierbare Version des Koinduktionsprinzips zum Beweis von Prozessgleichungen [23]. In [24] finden sich Ansätze zu einer koalgebraischen Modellierung des Ambientenkalküls [6].

2.2.2 Koalgebraische Modallogik

Neben der Klärung nach [50] verbliebener offener Fragen hinsichtlich der Hennessy-Milner-Eigenschaft allgemeiner koalgebraischer Modallogiken [57] besteht der Hauptbeitrag des Antragstellers zur Entwicklung der koalgebraischen Modallogik in der Begründung einer koalgebraisch basierten Theorie generischer Entscheidungsverfahren für Modallogiken, die zum einen tatsächlich in großer Allgemeinheit anwendbar ist und zum anderen auf akzeptable oder sogar scharfe obere Komplexitätsschranken führt (im Gegensatz zu einem Entscheidbarkeitsresultat in [49], das aufgrund starker Annahmen auf nur wenige Logiken anwendbar ist und dort typischerweise nur nicht-elementare obere Schranken liefert).

Der Beginn dieser Theorie ist der Beweis der (voraussetzungsfreien!) endlichen Modelleigenschaft für koalgebraische Modallogik; der aus der Konstruktion erwachsende Algorithmus liefert unter sehr schwachen Annahmen eine doppelt exponentielle obe-

re Komplexitätsschranke (genauer: NEXPTIME). Bereits dieses Einstiegsresultat hat erhebliche Aufmerksamkeit erregt [58]. Die obere Komplexitätsschranke lässt sich in vielen Fällen mittels flacher Modelle auf PSPACE verbessern; dies verlangt stärkere Annahmen entweder hinsichtlich der Axiomatisierung [61] oder auf semantischer Ebene [59]. Mit den genannten Methoden lassen sich bekannte scharfe obere Schranken u.a. für die Standardmodallogik K [37], probabilistische Modallogik [25, 38, 13], gradierte Modallogik [67], Koalitionslogik [51], und die Konditionallogik CK [47] nicht nur in einem allgemeinem Rahmen, sondern auch mit stark verschlankten Beweisen reproduzieren. Ferner ist in der Konferenzpräsentation von [61] eine zu diesem Zeitpunkt (d.h. kurz vor dem Erscheinen von [10]) noch knapp als neu anzusehende scharfe obere PSPACE-Schranke für Majoritätslogik [48] zu finden. In [62] wird mit Mitteln der mehrsortigen Koalgebra gezeigt, dass die Methode von [61] modular bezüglich Systemtypen ist, was auf PSPACE-Entscheidungsverfahren für Modallogiken über Systemen gemischten Verzweigungstyps führt. Die vielleicht typischste Klasse solcher Systeme sind Systeme mit sowohl nichtdeterministisch als auch probabilistisch verzweigenden Zuständen (s. Abschnitt 2.1.4), denen auf logischer Seite eine Mischung von Standardmodaloperatoren und probabilistischen Operatoren im Sinne einer mehrsortigen Modallogik entspricht.

In [58] und [63] wird bewiesen, dass die Klasse der koalgebraischen Modallogiken zusammenfällt mit der Klasse der in Rang 1 (d.h. mit homogener Schachtelungstiefe 1 für Modaloperatoren) axiomatisierbaren Modallogiken. Wie die obigen Beispiele zeigen, ist dies eine breite und wichtige Klasse; dennoch bleibt die Ausdehnung der erreichten Resultate auf allgemeinere Klassen von Logiken ein zentrales offenes Problem.

3 Ziele und Arbeitsprogramm

3.1 Ziele

Modallogiken verschiedenster Natur sind von wachsender Bedeutung für zentrale Gebiete der Informatik und künstlichen Intelligenz; hierbei spielen insbesondere Nichtstandardsemantiken (d.h. solche, die von der üblichen Kripke-Semantik abweichen) eine große Rolle (s. Abschnitt 2.1.1). Die Proliferation laufend neu entworfener Logiken ist hierbei keineswegs als unnütze Spielerei anzusehen, sondern ist vielmehr begründet in der Natur der Modallogik als anpassungsfähiges Ausdrucksmittel für sehr verschiedene, aber jeweils eng umrissene Anwendungsgebiete. Eine vereinheitlichende semantische und syntaktische Theorie von Modallogiken, die insbesondere generische und damit wiederverwendbare Algorithmen beinhaltet, stellt insofern einen entscheidenden Fortschritt dar.

Ein solches vereinheitlichendes Rahmenwerk stellt die jüngst entwickelte *koalgebraische Modallogik* dar. Die vom Antragsteller begründete hierauf basierende Theorie der generischen Algorithmen und Komplexitätsschranken für Modallogiken hat in ihrem gegenwärtigen Anwendbarkeitsbereich, dem der Rang-1-Logiken, einen zufriedenstellenden Stand erreicht (s. Abschnitt 2.2.2), der letztlich für im wesentlichen alle betreffenden Logiken den Beweis scharfer oberer Komplexitätsschranken erlaubt (der Beweis unterer Schranken durch Einbettung von Logiken bekannter Komplexität ist typischerweise leicht). Ziel des hier beantragten Projekts ist primär die Ausdehnung der Theorie

auf allgemeinere Klassen von Logiken. Im Fokus stehen hierbei zwei Grundtypen:

Kovarietale Modallogiken/Frame Conditions Die Beschränkung der bisherigen Theorie auf Rang 1 ist letztlich verursacht durch die gewählte Interpretation über der Klasse *aller* koalgebraischen Modelle, in Analogie zur Interpretation der klassischen Modallogik K über der Klasse aller Kripke-Rahmen. Dies schließt z.B. wichtige klassische Logiken wie $S4$ aus, die über durch *globale Rahmeneigenschaften* (*frame conditions*) bestimmten Modellklassen (hier: über transitiven und reflexiven Rahmen) interpretiert werden und durch Axiome außerhalb Rang 1 axiomatisiert sind (im Falle von $S4$ z.B. das Axiom $\Box a \rightarrow \Box \Box a$), während Rang-1-Logiken nur *lokale* Rahmeneigenschaften spezifizieren. Nichtklassische Beispiele von Axiomen außerhalb Rang 1 sind die Introspektionsaxiome der epistemischen Logik (s. etwa [72]) und der probabilistischen Modallogik [25] sowie verschiedene Postulate der Konditionallogik (s. z.B. [16]). Die koalgebraische Modellierung globaler Rahmeneigenschaften beruht auf Klassen von Koalgebren mit gewissen Abschlußeigenschaften, sogenannten Kovarietäten [56]. Zu erwarten ist, dass letztlich im wesentlichen *jede* Modallogik kovarietal in diesem Sinne ist; wir wollen hier das Schlagwort „kovarietale Modallogik“ aber als auf Logiken ohne Fixpunktoperatoren beschränkt verstehen.

Fixpunktlogiken Für monotone Modaloperatoren lassen sich stets bestimmte Fixpunkte definieren, die auf – für Zwecke der Spezifikation reaktiver Systeme entscheidende – Modaloperatoren temporaler Natur führen. Die einfachste Form dieser Fixpunktbildung ist die Iteration. Da die Iteration der üblichen Modaloperatoren \Box und \Diamond mit Kripke-Semantik gerade die Operatoren AG bzw. EG der Computation Tree Logic [8] liefert, bietet sich für koalgebraische Modallogiken mit Iteration das Schlagwort *koalgebraisches CTL* an. Die Instantiierung dieses Konzepts auf nicht-normale (d.h. das K -Axiom verletzende) Logiken wie gradierete Modallogik, probabilistische Modallogik oder Koalitionslogik liefert Modaloperatoren, die etwa über die Existenz erfüllender Unterstrukturen mit gegebener Mindestverzweigungsbreite oder die Fähigkeit einer Koalition von Agenten, bestimmte Invarianten zu erzwingen, reden. (Im Falle der Koalitionslogik wird es so insbesondere möglich, modallogisch über Gewinnstrategien in Spielen mit unbeschränkter Anzahl von Zügen zu sprechen.)

Zur Illustration der Breite des Gebiets gibt Tabelle 1 einen Überblick über eine Auswahl von Modallogiken aus den gerade genannten Klassen, aus der bereits mit den existierenden Methoden behandelbaren Klasse der Rang-1-Logiken, und weitere. (Nicht nur erhebt die Tabelle keinen Anspruch auf Vollständigkeit, sondern die Unvollständigkeit bzw. Fortsetzbarkeit der Liste ist eine der Motivationen für das generische Rahmenwerk.)

Ziel hinsichtlich kovarietaler Modallogiken ist es, Klassen von Axiomen und Strukturen zu identifizieren, die Entscheidungsprozeduren in PSPACE erlauben (uneingeschränkt allgemeine Resultate sind insofern nicht zu erwarten, als bereits im klassischen Fall nicht jede Modallogik außerhalb Rang 1 in PSPACE entscheidbar ist), in Verallgemeinerung klassischer Resultate für T , $S4$, $S5_n$ und weitere normale Modallogiken [37, 20]. Hieran werden sich, ähnlich wie im Falle der Rang-1-Logiken [61, 62],

	Komplexität	Anmerkungen
Rang-1-Logiken		
K, KD	PSPACE [37]	
E, M	NP [72]	
Koalitionslogik	PSPACE [51]	
Gradierte Modallogik	PSPACE [67]	
Majoritätslogik	PSPACE [10, 61]	
Presburger-Modallogik	PSPACE [10]	Subsumiert gradierte ML und Majoritätslogik
Probabilistische Modallogik	PSPACE [13, 61]	In [61] erstmals Tableau-Alg.
Konditionallogiken $CK\{+Id\}$	PSPACE [47]	Sehr kurzer Beweis in [59]
Deontische Logik für Dilemmata [17]	?	
Kovarietale Modallogiken (Frame Conditions)		
$S4, T$	PSPACE [37]	
$S5$	NP [37]	
$S5_n$	PSPACE [20]	
Epistemische Logiken mit Introspektion	PSPACE [72]	
Rollenhierarchien und Transitivität	EXPTIME [20]	
Gradierte ML mit Rollen-hierarchien und Transitivität	Unentscheidbar [28]	
Gradierte ML mit Transitivität	?	
Probabilistische ML mit Introspektion [25]	?	
Default-Logik mit Skyrms-Axiomen [66]	?	
Koalgebraisches CTL (Iteration)		
CTL	EXPTIME [11]	
Gradierte ML mit Iteration	EXPTIME [32]	Aber noch kein vollständiger Kalkül und kein Tableau-Alg.
Koalitionslogik mit Iteration	?	
Probabilistische ML mit Iteration	?	
Weitere Logiken		
μ -Kalkül	EXPTIME [12]	
Gradierter μ -Kalkül	EXPTIME [32]	Subsumiert gradierte ML mit Iteration

Tabelle 1: Eine Auswahl von Modallogiken

beweistheoretische Resultate wie etwa schwache Formen der Unterformeleigenschaft anschließen.

Für koalgebraisches CTL ist zunächst eine vollständige Axiomatisierung zu finden; Hauptziel ist sodann der Beweis einer allgemeinen oberen Schranke EXPTIME, in Verallgemeinerung der EXPTIME-Vollständigkeit von CTL [11].

Im Anschluss an diese Resultate sind experimentelle Implementierungen der gewonnenen Entscheidungsprozeduren vorgesehen, die in unoptimierter Form wegen der zu erwartenden exponentiellen Laufzeit zunächst nur auf relativ kleine Formeln anwendbar sein werden. Die eigentliche Bedeutung der generischen Algorithmen liegt in den entsprechenden oberen Komplexitätsschranken, die als Indiz für die Machbarkeit vollautomatischer modaler Beweiser anzusehen sind — z.B. ist eine in PSPACE entscheidbare Modallogik nicht komplexer als Standardmodallogiken wie K , für die effektive modale Beweiser existieren (s. Abschnitt 2.1.3). Geplant sind daher weiterhin experimentelle heuristische Optimierungen der generischen Algorithmen als Vorstudie für ein Folgeprojekt, das die Implementierung eines leistungsfähigen generischen modalen Beweiswerkzeugs beinhaltet.

3.2 Arbeitsprogramm

Im folgenden ist eine Arbeitsplanung in Form von acht Arbeitspaketen (AP) angegeben; Abhängigkeiten zwischen den Arbeitspaketen ergeben sich aus den jeweiligen Beschreibungen. Ein detaillierter Zeitplan zu den Arbeitspaketen findet sich in Tabelle 2.

3.2.1 AP1: Infrastruktur für ein generisches Beweiswerkzeug

Die unter AP3–7 zu entwickelnden generischen Algorithmen und ihre konkreten Instanzen werden im Rahmen der jeweiligen Arbeitspakete fortlaufend experimentell implementiert. Hierzu wird Infrastruktur in Gestalt allgemeiner Formate für die Repräsentation von Modellen (für die semantikbasierten Algorithmen aus AP4) und von Tableau-Regeln (AP3 und AP6) sowie einer geeigneten generischen Eingabesyntax und eines entsprechenden Parsierungswerkzeugs bereitgestellt. Hierauf aufbauend werden zunächst die bereits vorliegenden generischen und spezifischen Algorithmen aus [59, 61] implementiert. Das entstehende generische modale Beweiswerkzeug wird modular gegenüber Kombination von Modallogiken im Sinne von [62] ausgelegt. Es wird erwartet, dass die Effizienz des unoptimierten Werkzeugs für den Beweis oder die Widerlegung typischer von Hand eingegebener Einzelformeln ausreicht.

Sowohl im Sinne eines Rapid Prototyping als auch zur besseren Unterstützung der anvisierten Generizität bietet sich als Implementierungssprache die funktionale Programmiersprache Haskell [52] an, mit deren Verwendung am DFKI-Labor Bremen durch die Implementierung des Bremer heterogenen Werkzeugsatzes [43] (ca. 80.000 Zeilen Code) umfangreiche Erfahrungen vorliegen.

3.2.2 AP2: Endliche Modelle für kovarietale Modallogiken

Als Vorstudie für die in AP3 und AP4 benötigten Konstruktionen flacher Modelle wird zunächst eine filtrationsbasierte Konstruktion endlicher Modelle für geeignete kovarietale Modallogiken gesucht (s. Abschnitt 3.1). Die Vorbilder hierfür liefern die endliche

Modellkonstruktion für koalgebraische Modallogiken (d.h. in Rang 1) aus [58] sowie entsprechende Konstruktionen für normale Modallogiken außerhalb Rang 1 (S4, S5 etc.) [20]. Zu erwartende Folgeresultate (für solche Logiken, die die Anwendungsvoraussetzungen des Verfahrens erfüllen) sind neben der endlichen Modelleigenschaft

- schwache Vollständigkeit (d.h. jede gültige Formel ist herleitbar)
- Entscheidbarkeit des Erfüllbarkeitsproblems
- NEXPTIME als obere Komplexitätsschranke.

Die obere Komplexitätsschranke hängt hierbei voraussichtlich nur von recht schwachen Bedingungen ab. Sie wird sich in vielen (aber bei weitem nicht allen) Fällen verbessern lassen; dies ist Gegenstand von AP3–5.

3.2.3 AP3: Generische Tableau-Algorithmen für kovarietale Modallogiken

Der tableau-orientierte generische PSPACE-Algorithmus für Erfüllbarkeit in Rang-1-Logiken [61] basiert im wesentlichen auf der Beobachtung, dass sich für viele Logiken die Axiomatisierung unter einer geeigneten Form von Resolution abschließen lässt, bzw. das Ergebnis dieses Abschlussprozesses unter Kontrolle gebracht werden kann. Dieses Verfahren wird auf kovarietale Modallogiken (s. Abschnitt 3.1) erweitert und im Rahmenwerk aus AP1 implementiert. Hierfür sind geeignete syntaktische Vorbedingungen zu identifizieren, die über bloße Restriktionen der Axiomenformate vermutlich deutlich hinausgehen werden. (Bereits recht harmlos wirkende Axiomenformate können zu hoher Komplexität führen. So ist z.B. in Deskriptionslogiken die Kombination von Rollenhierarchien, transitiven Rollen und gradierten Modaloperatoren bereits unentscheidbar [28]). Eine realistische Perspektive besteht ferner in der Ausdehnung der in [20] detailliert beschriebenen Tableaumethoden für insbesondere S4 und S5_n auf den koalgebraischen Fall; diese Methoden beruhen auf zum Teil zu AP2 analogen Modellkonstruktionsprinzipien. Wie bereits für S4 wird eine allgemeine Analyse von Zykeln notwendig werden, von denen man allerdings hoffen kann, dass sie in den Modellkonstruktionen automatisch abfallen.

3.2.4 AP4: Generische semantikbasierte PSPACE-Algorithmen

Parallel zu AP3 wird der semantikbasierte Erfüllbarkeitsalgorithmus aus [59] auf kovarietale Modallogiken (s. Abschnitt 3.1) ausgedehnt und im Rahmenwerk aus AP1 implementiert, ggf. unter Anpassung des Modellrepräsentationsformats. Der Gesamtansatz basiert auf der Verklebung kleiner lokaler Teilmodelle zu einem flachen globalen Modell, ein für den Spezialfall der probabilistischen Modallogik in [13] zum Teil implizites Verfahren. Der Verklebungsprozess wird im allgemeinen Fall hierbei erheblich komplizierter als im Rang-1-Fall sein, da Modelle nicht mehr rein lokal konstruiert werden können, sondern gleichzeitig globale Rahmeneigenschaften sicherzustellen sind (vgl. Abschnitt 3.1). Der Mehrwert dieses alternativen generischen Algorithmen-Schemas liegt insbesondere in seiner oft extrem einfachen Anwendbarkeit, etwa im Falle verschiedener Konditionallogiken [59], sowie in der Tatsache, dass man bei Beschränkung der Schachtelungstiefe von Modaloperatoren sogar eine obere Komplexitätsschranke NP erhält.

3.2.5 AP5: Instanzen der generischen PSPACE-Algorithmen

Als Anwendungsbeispiele für sowohl die semantikbasierten als auch die tableaubasierten PSPACE-Algorithmen aus AP3 und AP4 sind etwa gradierte Modaloperatoren mit Transitivitätsrestriktionen (aber ohne Rollenhierarchien, s.o.) [27], bestimmte Default-Logiken [66] sowie probabilistische Modallogiken mit Introspektion [25] anvisiert, letztere z.B. formuliert als Axiom (*Frame Condition*)

$$L_{\alpha}^i \phi \rightarrow L_1^i L_{\alpha}^i \phi,$$

zu lesen „wenn Agent i ϕ mit Wahrscheinlichkeit mindestens α glaubt, dann weiß er mit Sicherheit (d.h. Wahrscheinlichkeit 1), dass er dies glaubt“. Die Instantiierung der tableau-basierten Methode verlangt die Bildung geeignet abgeschlossener Axiomatisierungen der jeweiligen Logiken, während die semantikbasierten Methode den Beweis lokaler polynomieller Modelleigenschaften voraussetzt. Die Spezialisierung der in AP3 und AP4 implementierten Algorithmen besteht dann in einer geeigneten Instantiierung der allgemeinen Regel- bzw. Modellrepräsentation. In allen genannten Fällen und voraussichtlich weiteren führt die Instantiierung der generischen Algorithmen im Erfolgsfall zu bisher unbekanntem PSPACE-Schranken, bzw. NP-Schranken bei Beschränkung der Schachtelungstiefe und Verwendung der Methode aus AP4.

3.2.6 AP6: Pseudomodelle und EXPTIME-Algorithmen für koalgebraisches CTL

Eine der verzweigenden Temporallogik CTL analoge temporale Erweiterung der koalgebraischen Modallogik entsteht durch Hinzunahme durch eine einfache Fixpunktgleichung bestimmter iterativer Modaloperatoren (s. Abschnitt 3.1). Für die so entstehenden Logiken sind bis auf sehr eng begrenzte Spezialfälle (CTL selbst [11] und die Erweiterung um gradierte Operatoren [32]) sowohl die Entscheidbarkeit des Erfüllbarkeitsproblems und dessen Komplexität (in den erwähnten Spezialfällen EXPTIME) als auch bereits die Frage der vollständigen Axiomatisierung noch völlig offen. Die besagte Fixpunktgleichung induziert in naheliegender Weise eine Beweisregel für iterative Operatoren; es wird vermutet, dass diese in Verbindung mit einer Hintergrundaxiomatisierung der vorliegenden koalgebraischen Modallogik einen vollständigen Kalkül liefert.

Zum Beispiel liefert Instanziierung der Iteration auf den klassischen Operator \square gerade die übliche Temporallogik bzw. bei zusätzlicher Indizierung von Modaloperatoren (*multimodale Logik* oder *Hennesy-Milner-Logik*) die propositionale dynamische Logik, und aus der besagten Regel gewinnt man leicht das (für diesen Fall vollständige) Segerbergsche Induktionsaxiom [64]. Es ist aber fraglich, ob sich der Vollständigkeitsbeweis im allgemeinen Fall über endliche kanonische Modelle wie im klassischen Fall [4] führen lässt; vielversprechender scheint der Ansatz, die Vollständigkeit wie im Falle von CTL (das in Gestalt der Pfadoperatoren AG und EG Iteration nicht nur von \square , sondern auch von \diamond beinhaltet) simultan mit der Entscheidbarkeit in EXPTIME durch eine Pseudomodellkonstruktion [11] abzuhandeln. Aus dieser Konstruktion erwachsen zwei Entscheidungsverfahren für Erfüllbarkeit, insbesondere ein Tableau-Verfahren, von dem in der Praxis höhere Effizienz zu erwarten ist. Beide Verfahren werden im Rahmenwerk aus AP1 experimentell implementiert.

Quartal	1	2	3	4	5	6	7	8	9	10	11	12
AP1	2	1										
AP2	1	2	1									
AP3			2	3	1							
AP4						3	2					
AP5					2		1	1				
AP6								2	3	1		
AP7										2	1	
AP8											2	3

Tabelle 2: Zeitplan. Angegeben ist der Aufwand pro Quartal in Personenmonaten.

3.2.7 AP7: Instanzen für koalgebraisches CTL

Die generische Temporallogik („koalgebraisches CTL“) und die in AP6 entwickelten und implementierten zugehörigen generischen Entscheidungsverfahren werden durch Beweis der sich notwendigerweise ergebenden Zusatzannahmen und Implementierung geeigneter Regel- und Modellrepräsentationen für konkrete Modallogiken instanziiert. Interessante Anwendungen ergeben sich insbesondere in der Koalitionslogik [51], in der iterierte Operatoren Aussagen der Natur „Koalition C hat eine Strategie, im verbleibenden Spielverlauf Invariante ϕ zu erzwingen“ erlauben, und in der Analyse probabilistischer Systeme mittels probabilistischer Modallogik, die nunmehr um eine temporale Komponente erweitert wird (die zwar von PCTL [22] wesentlich abweicht, aber eigene Berechtigung hat). Anwendung auf gradierte Modallogik ergibt ferner iterierte Operatoren, die die Existenz von erfüllenden Unterbäumen mit bestimmter Mindestverzweigungsbreite zusichern; zwar folgt Entscheidbarkeit dieser Logik in EXPTIME bereits aus den Resultaten über den gradierten μ -Kalkül [32], man erhält aber einen neuen tableaubasierten (statt automatenbasierten) Algorithmus, der voraussichtlich besser zugänglich für heuristische Optimierungen ist (s. AP8).

3.2.8 AP8: Optimierungsvorstudie

Schließlich soll eruiert werden, ob sich das generische Beweiswerkzeug aus AP1 in Zukunft auf einen für den Umgang mit modalen Wissensbasen geeigneten Effizienzgrad optimieren lässt; Fernziel ist ein generischer modaler Beweiser, der in seiner Effizienz mit spezialisierten Deskriptionslogikbeweisern (s. Abschnitt 2.1.3) zumindest vergleichbar ist. Hierzu werden Vorstudien über geeignete Optimierungsstrategien durchgeführt; als Ausgangspunkt bieten sich die in [27, 26, 68, 65] beschriebenen Optimierungen von Tableausystemen an. Diese sind zu ergänzen durch benutzersteuerbare logikspezifische Heuristiken wie z.B. Gewichtung von Tableau-Regeln; für gradierte und probabilistische Logiken spielen ferner lineare Optimierungsprobleme eine Rolle, die durch Ankopplung existierender Software behandelt werden können. Der Effekt der verschiedenen Optimierungen wird anhand von Benchmark-Formelserien evaluiert, die analog zu entsprechenden Sammlungen für Standardmodallogiken (z.B. [41]) entwickelt werden.

4 Beantragte Mittel

4.1 Personalkosten

Für die Laufzeit des Projekts werden Mittel für eine volle BAT Ila-Stelle eines wissenschaftlichen Mitarbeiters beantragt. Zusätzlich werden, ebenfalls für die Laufzeit des Projekts, Mittel für eine studentische Hilfskraft für Programmieraufgaben im Umfang von 40 Stunden im Monat beantragt.

4.4 Reisen

Für den Antragsteller sowie den wissenschaftlichen Mitarbeiter werden für die anfallende Forschungsarbeit und die Präsentation der Projektergebnisse pro Jahr jeweils zwei Reisen zu den unter 5.2 genannten Kooperationspartnern und zu internationalen Tagungen veranschlagt (einschlägige Tagungen sind etwa Foundations of Software Science and Computation Structures (FOSSACS), Tools and Algorithms for the Construction and Analysis of Systems (TACAS), Coalgebraic Methods in Computer Science (CMCS), Workshop on Algebraic Development Techniques (WADT), Algebra and Coalgebra in Computer Science (CALCO), Automata, Languages and Programming (ICALP), Advances in Modal Logic (AiML), Concurrency Theory (CONCUR), Logic in Computer Science (LICS), Principles of Knowledge Representation and Reasoning (KR)), für die jeweils EUR 1.300,- angesetzt werden (typische Rechnung: Tagungsgebühren EUR 400,-, Flug EUR 400,-, Übernachtungs- und Verpflegungskosten 5 Tage à EUR 100). Dementsprechend werden für die ersten zwei Jahre insgesamt

$$\text{EUR } 1.300 \times 2 \times 2 \times 2 = \text{EUR } 10.400,-$$

und für das dritte Jahr

$$\text{EUR } 1.300 \times 2 \times 2 = \text{EUR } 5.200,-$$

an Reisekosten beantragt.

5 Voraussetzungen für die Durchführung des Vorhabens

5.1 Zusammensetzung der Arbeitsgruppe

Der Antragsteller ist als Senior Researcher in der Forschungsgruppe Sichere Kognitive Systeme am DFKI-Labor Bremen (Leitung: Prof. Dr. Bernd Krieg-Brückner) tätig. In derselben Forschungsgruppe arbeiten PD Dr. Till Mossakowski, PD Dr. Christoph Lüth, Dipl. Inf. Daniel Hausmann und Dipl. Inf. Dennis Walter an verwandten Themen und sind insbesondere im Bereich der koalgebraischen Methoden in der Informatik international ausgewiesen.

5.2 Zusammenarbeit mit anderen Wissenschaftlern

1. Dr. Dirk Pattinson, Imperial College London, UK (Koalgebra und Modallogik)
2. Prof. Dr. Joseph Y. Halpern, Cornell University, USA (Modallogik in der KI)

5.3 Arbeiten im Ausland und Kooperation mit Partnern im Ausland

Im Rahmen des Vorhabens ist eine Zusammenarbeit mit den unter 5.2 genannten Kooperationspartnern geplant.

5.4 Apparative Ausstattung

Das DFKI-Labor Bremen stellt für den unter 4.1 beantragten Mitarbeiter einen leistungsfähigen Arbeitsplatzrechner zur Verfügung, der in das vorhandene Rechnernetz eingebunden wird.

5.5 Laufende Mittel für Sachausgaben

Laufende Kosten für Schreibbedarf etc. werden im Rahmen vorhandener Haushaltsmittel des DFKI-Labors Bremen zur Verfügung gestellt.

6 Erklärungen

Ein Antrag auf Finanzierung dieses Vorhabens wurde bei keiner anderen Stelle eingereicht. Wenn ich einen solchen Antrag stelle, werde ich die Deutsche Forschungsgemeinschaft unverzüglich benachrichtigen.

7 Unterschrift

Bremen, den

Lutz Schröder

8 Verzeichnis der Anlagen

- Lebenslauf
- Ausgewählte Publikationen: [57, 58, 61, 63, 62]
- Verzeichnis der antragsrelevanten wissenschaftlichen Veröffentlichungen

Literatur

- [1] F. Bartels. Generalised coinduction. *Math. Struct. Comput. Sci.*, 13:321–348, 2003.
- [2] B. Beckert and R. Goré. System description: leanK 2.0. In C. Kirchner and H. Kirchner, editors, *Automated Deduction, CADE 98*, volume 1421 of *Lect. Notes Artificial Intelligence*, pages 51–55. Springer, 1998.
- [3] M. Bidoit and P. D. Mosses. *CASL User Manual*, volume 2900 of *Lect. Notes Comput. Sci.* Springer, 2004.
- [4] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge, 2001.
- [5] P. A. Bonatti, C. Lutz, A. Murano, and M. Y. Vardi. The complexity of enriched μ -calculi. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, editors, *Automata, Languages and Programming, 33rd International Colloquium, ICALP 06*, volume 4052 of *Lect. Notes Comput. Sci.*, pages 540–551. Springer, 2006.
- [6] L. Cardelli and A. Gordon. Mobile ambients. *Theoret. Comput. Sci.*, 240:177–213, 2000.
- [7] C. Cirstea and D. Pattinson. Modular construction of modal logics. *Theoret. Comput. Sci.* To appear. Earlier version in P. Gardner, N. Yoshida, editors, *Concurrency Theory, CONCUR 04*, vol. 3170 of *Lect. Notes Comput. Sci.*, pp. 258–275, Springer, 2004.
- [8] E. M. Clarke and E. A. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In D. Kozen, editor, *Logics of Programs*, volume 131 of *Lect. Notes Comput. Sci.*, pages 52–71. Springer, 1981.
- [9] H. De Nivelle, R. A. Schmidt, and U. Hustadt. Resolution-based methods for modal logics. *Logic J. IGPL*, 8:265–292, 2000.
- [10] S. Demri and D. Lugiez. Presburger modal logic is only PSPACE-complete. In U. Furbach and N. Shankar, editors, *Automated Reasoning, IJCAR 06*, volume 4130 of *Lect. Notes Artificial Intelligence*, pages 541–556. Springer, 2006.
- [11] E. A. Emerson and J. Y. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. *J. Comput. System Sci.*, 30:1–24, 1985.
- [12] E. A. Emerson and C. S. Jutla. The complexity of tree automata and logics of programs. *SIAM J. Comput.*, 29:132–158, 1999.
- [13] R. Fagin and J. Y. Halpern. Reasoning about knowledge and probability. *J. ACM*, 41:340–367, 1994.
- [14] K. Fine. In so many possible worlds. *Notre Dame J. Formal Logic*, 13:516–520, 1972.
- [15] M. J. Fischer and R. E. Ladner. Propositional dynamic logic of regular programs. *J. Comput. System Sci.*, 18:194–211, 1979.
- [16] N. Friedman and J. Y. Halpern. On the complexity of conditional logics. In J. Doyle, E. Sandewall, and P. Torasso, editors, *Principles of Knowledge Representation and Reasoning, KR 94*, pages 202–213. Morgan Kaufmann, 1994.
- [17] L. Goble. A proposal for dealing with deontic dilemmas. In *Deontic Logic in Computer Science*, volume 3065 of *Lect. Notes Artificial Intelligence*, pages 74–113. Springer, 2004.

- [18] V. Haarslev and R. Möller. RACER system description. In R. Goré, A. Leitsch, and T. Nipkow, editors, *International Joint Conference on Automated Reasoning, IJCAR 2001*, volume 2083 of *Lect. Notes Comput. Sci.*, pages 701–705. Springer, 2001.
- [19] J. Y. Halpern. *Reasoning about Uncertainty*. MIT Press, 2003.
- [20] J. Y. Halpern and Y. O. Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54:319–379, 1992.
- [21] J. Y. Halpern and R. Pucella. Modeling adversaries in a logic for security protocol analysis. In A. E. Abdallah, P. Ryan, and S. Schneider, editors, *Formal Aspects of Security, FASEc 02*, volume 2629 of *Lect. Notes Comput. Sci.*, pages 115–132. Springer, 2002.
- [22] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6:512–535, 1994.
- [23] D. Hausmann, T. Mossakowski, and L. Schröder. Iterative circular coinduction for CoCASL in Isabelle/HOL. In M. Cerioli, editor, *Fundamental Approaches to Software Engineering, FASE 05*, volume 3442 of *Lect. Notes Comput. Sci.*, pages 341–356. Springer, 2005.
- [24] D. Hausmann, T. Mossakowski, and L. Schröder. A coalgebraic approach to the semantics of the ambient calculus. *Theoretical Computer Science*, 366:121–143, 2006.
- [25] A. Heifetz and P. Mongin. Probabilistic logic for type spaces. *Games and Economic Behavior*, 35:31–53, 2001.
- [26] I. Horrocks. Implementation and optimisation techniques. In F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. F. Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementation, and Applications*, chapter 9, pages 306–346. Cambridge University Press, 2003.
- [27] I. Horrocks and P. F. Patel-Schneider. Optimising description logic subsumption. *J. Logic Comput.*, 9:267–293, 1999.
- [28] I. Horrocks, U. Sattler, and S. Tobies. Practical reasoning for expressive description logics. In H. G. an D. McAllester and A. Voronkov, editors, *Logic for Programming and Automated Reasoning, LPAR 99*, volume 1705 of *Lect. Notes Comput. Sci.*, pages 161–180. Springer, 1999.
- [29] B. Jacobs. Towards a duality result in the modal logic of coalgebras. In B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors, *Coalgebraic Methods in Computer Science, CMCS 00*, volume 33 of *Electron. Notes Theoret. Comput. Sci.* Elsevier, 2000.
- [30] D. Kozen. Results on the propositional μ -calculus. *Theoret. Comput. Sci.*, 27:333–354, 1983.
- [31] O. Kullmann. The SAT 2005 solver competition on random instances. *J. Satisfiability, Boolean Modeling and Computation*, 2:61–102, 2006.
- [32] O. Kupferman, U. Sattler, and M. Y. Vardi. The complexity of the graded μ -calculus. In A. Voronkov, editor, *Automated Deduction, CADE 02*, volume 2392 of *Lect. Notes Comput. Sci.*, pages 423–437. Springer, 2002.
- [33] C. Kupke, A. Kurz, and D. Pattinson. Algebraic semantics for coalgebraic logics. In J. Adámek and S. Milius, editors, *Coalgebraic Methods in Computer Science*, volume 106 of *Electron. Notes Theoret. Comput. Sci.*, pages 219–241. Elsevier, 2004.

- [34] C. Kupke, A. Kurz, and D. Pattinson. Ultrafilter extensions for coalgebras. In J. Fiadeiro, N. Harman, M. Roggenbach, and J. Rutten, editors, *Algebra and Coalgebra in Computer Science*, volume 3629 of *Lect. Notes Comput. Sci.*, pages 263–277. Springer, 2005.
- [35] C. Kupke and Y. Venema. Closure properties of coalgebra automata. In P. Panangaden, editor, *Logic in Computer Science, LICS 05*, pages 199–208. IEEE, 2005.
- [36] A. Kurz. Specifying coalgebras with modal logic. *Theoret. Comput. Sci.*, 260:119–138, 2001.
- [37] R. E. Ladner. The computational complexity of provability in systems of modal propositional logic. *SIAM J. Comput.*, 6, 1977.
- [38] K. Larsen and A. Skou. Bisimulation through probabilistic testing. *Inform. Comput.*, 94:1–28, 1991.
- [39] A. Lomuscio and D. Nute, editors. *Deontic Logic in Computer Science, DEON 2004*, volume 3065 of *Lect. Notes Comput. Sci.* Springer, 2004.
- [40] Z. Manna and A. Pnueli. The modal logic of programs. In H. A. Maurer, editor, *Automata, Languages and Programming, ICALP 79*, volume 71 of *Lect. Notes Comput. Sci.*, pages 385–409. Springer, 1979.
- [41] F. Massacci and F. Donini. Design and results of tancs-2000 non-classical (modal) systems comparison. In R. Dyckhoff, editor, *Automated Reasoning with Analytic Tableaux and Related Methods*, volume 1847 of *Lect. Notes Comput. Sci.*, pages 52–56. Springer, 2000.
- [42] L. Moss. Coalgebraic logic. *Ann. Pure Appl. Logic*, 96:277–317, 1999.
- [43] T. Mossakowski, C. Maeder, and K. Lüttich. The Heterogeneous Tool Set. In O. Grumberg and M. Huth, editors, *Tools and Algorithms for the Construction and Analysis of Systems, TACAS 07*, volume 4424 of *Lect. Notes Comput. Sci.*, pages 519–522. Springer, 2007.
- [44] T. Mossakowski, L. Schröder, M. Roggenbach, and H. Reichel. Algebraic-coalgebraic specification in CoCASL. *J. Logic Algebraic Programming*, 67:146–197, 2006.
- [45] P. D. Mosses, editor. *CASL Reference Manual*, volume 2960 of *Lect. Notes Comput. Sci.* Springer, 2004.
- [46] H. J. Ohlbach and J. Koehler. Modal logics, description logics and arithmetic reasoning. *Artif. Intell.*, 109:1–31, 1999.
- [47] N. Olivetti and C. Schwind. A calculus and complexity bound for minimal conditional logic. In A. Restivo, S. R. D. Rocca, and L. Roversi, editors, *Theoretical Computer Science, ICTCS 01*, volume 2202 of *Lect. Notes Comput. Sci.*, pages 384–404. Springer, 2001.
- [48] E. Pacuit and S. Salame. Majority logic. In D. Dubois, C. Welty, and M.-A. Williams, editors, *Principles of Knowledge Representation and Reasoning, KR 04*, pages 598–605. AAAI Press, 2004.
- [49] D. Pattinson. Coalgebraic modal logic: Soundness, completeness and decidability of local consequence. *Theoret. Comput. Sci.*, 309:177–193, 2003.
- [50] D. Pattinson. Expressive logics for coalgebras via terminal sequence induction. *Notre Dame J. Formal Logic*, 45:19–33, 2004.

- [51] M. Pauly. A modal logic for coalitional power in games. *J. Logic Comput.*, 12:149–166, 2002.
- [52] S. Peyton-Jones, editor. *Haskell 98 Language and Libraries — The Revised Report*. Cambridge, 2003. also: *J. Funct. Programming* **13** (2003).
- [53] A. Pnueli. The temporal logic of programs. In *Foundations of Computer Science, FOCS 77*, pages 46–57. IEEE, 1977.
- [54] V. R. Pratt. Semantical considerations on floyd-hoare logic. In *Foundations of Computer Science, FOCS 76*, pages 109–121. IEEE, 1976.
- [55] M. RöBiger. Coalgebras and modal logic. In B. Jacobs, L. Moss, H. Reichel, and J. Rutten, editors, *Coalgebraic Methods in Computer Science*, volume 33 of *Electron. Notes Theoret. Comput. Sci.* Elsevier, 2000.
- [56] J. Rutten. Universal coalgebra: A theory of systems. *Theoret. Comput. Sci.*, 249:3–80, 2000.
- [57] L. Schröder. Expressivity of coalgebraic modal logic: the limits and beyond. *Theoret. Comput. Sci.* (FOSSACS 05 special issue). To appear. Earlier version in V. Sassone, editor, *Foundations of Software Science And Computation Structures, FOSSACS 05*, vol. 3441 of *Lect. Notes Comput. Sci.*, pp. 440–454, Springer, 2005.
- [58] L. Schröder. A finite model construction for coalgebraic modal logic. *J. Logic Algebraic Programming* (FOSSACS 06 special issue). To appear. Earlier version in L. Aceto and A. Ingólfssdóttir, editors, *Foundations Of Software Science And Computation Structures, FOSSACS 06*, vol. 3921 of *Lect. Notes Comput. Sci.*, Springer, 2006, pp. 157–171. EATCS Best Paper Award at ETAPS 06.
- [59] L. Schröder. A semantic PSPACE criterion for rank 1 modal logic. Unpublished manuscript.
- [60] L. Schröder and T. Mossakowski. Coalgebraic modal logic in CoCASL. In *Recent Developments in Algebraic Development Techniques, 18th International Workshop, WADT 06*, volume 4409 of *Lect. Notes Comput. Sci.*, pages 128–142. Springer, 2007.
- [61] L. Schröder and D. Pattinson. PSPACE reasoning for rank-1 modal logics. In R. Alur, editor, *Logic in Computer Science, LICS 06*, pages 231–240. IEEE, 2006. Presentation slides available under www.informatik.uni-bremen.de/~lschrode/slides/rank1pspace.pdf.
- [62] L. Schröder and D. Pattinson. Modular algorithms for heterogeneous modal logics. In L. Age, A. Tarlecki, and C. Cachin, editors, *Automata, Languages and Programming, ICALP 07*, *Lect. Notes Comput. Sci.* Springer, 2007. To appear.
- [63] L. Schröder and D. Pattinson. Rank-1 modal logics are coalgebraic. In W. Thomas and P. Weil, editors, *Theoretical Aspects of Computer Science, STACS 07*, volume 4393 of *Lect. Notes Comput. Sci.*, pages 573–585. Springer, 2007.
- [64] K. Segerberg. A completeness theorem in the modal logic of programs. *Notices of the AMS*, 24:A–552, 1977.
- [65] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical OWL-DL reasoner. *J. Web Semantics*. To appear.

- [66] B. Skyrms. Higher order degrees of belief. In D. H. Mellor, editor, *Prospects for Pragmatism: Essays in Honor of F. P. Ramsey*. Cambridge University Press, 1980.
- [67] S. Tobies. PSPACE reasoning for graded modal logics. *J. Logic Comput.*, 11:85–106, 2001.
- [68] D. Tsarkov and I. Horrocks. Ordering heuristics for description logic reasoning. In *Int. Joint Conf. on Artificial Intelligence, IJCAI 2005*, pages 609–61, 2005.
- [69] D. Tsarkov and I. Horrocks. FaCT++ description logic reasoner: System description. In *Int. Joint Conf. on Automated Reasoning, IJCAR 2006*, volume 4130 of *Lecture Notes in Artificial Intelligence*, pages 292–297. Springer, 2006.
- [70] W.-J. van den Heuvel and H. Weigand. Coordinating web-service enabled business transactions with contracts. In J. Eder and M. Missikoff, editors, *Advanced Information Systems Engineering, CAiSE 2003*, volume 2681 of *Lect. Notes Comput. Sci.*, pages 568–583. Springer, 2003.
- [71] W. van der Hoek and J.-J. Meyer. Graded modalities in epistemic logic. In A. Nerode, editor, *Logical Foundations of Computer Science*, volume 620 of *Lect. Notes Comput. Sci.*, pages 503–514. Springer, 1992.
- [72] M. Vardi. On the complexity of epistemic reasoning. In R. Parikh, editor, *Logic in Computer Science*, pages 243–251. IEEE, 1989.
- [73] Y. Venema. Automata and fixed point logics: a coalgebraic perspective. *Inform. Comput.* To appear. Earlier version in J. Adámek and S. Milius, editors, *Coalgebraic Methods in Computer Science, CMCS 04*, vol. 106 of *Electron. Notes Theoret. Comput. Sci.*, pp. 355–375.
- [74] G. H. von Wright. Deontic logic. *Mind*, 60:1–15, 1951.
- [75] W. Wobcke. Belief revision, conditional logic and nonmonotonic reasoning. *Notre Dame J. Formal Logic*, 36:55–102, 1995.