Optimised Modal Translation and Resolution

Dissertation

zur Erlangung des Grades Doktor der
Ingenieurwissenschaften (Dr.-Ing.) der Technischen
Fakultät der Universität des Saarlandes

von

Renate A. Schmidt

Saarbrücken, 1997

Dekan:	Prof. Dr. Alexander Koch
Gutachter:	Priv.-Doz. Dr. Hans Jürgen Ohlbach
	Prof. Dr. Harald Ganzinger
Tag des Kolloquiums:	3. November 1997

Abstract

This thesis studies the optimised functional translation of propositional modal logics to first-order logic, and first-order resolution as a means for realising modal reasoning. The optimised functional translation maps modal logics to a lattice of clausal logics, called path logics. The general apparatus of inference for path logics is theory resolution. We show that satisfiability in basic path logic and certain extensions can be decided by resolution and condensing without requiring additional refinement strategies. We propose an improved theory unification algorithm for *S4*, and we present a calculus of ordered E-resolution with normalisation. We show also that some essentially second-order modal logics convert to path logics, which can be exploited for accomodating inference for modal logics with numerical quantifiers in a calculus of resolution and simple arithmetic.

Zusammenfassung

Diese Arbeit untersucht die optimierte funktionale Übersetzung von modalen Aussagenlogiken in die Prädikatenlogik erster Stufe und deren Behandlung durch Resolutionsverfahren. Die optimierte funktionale Übersetzung bildet Modallogiken in Klausellogiken, genannt Pfadlogiken, ab. Der allgemeine Inferenzformalismus für Pfadlogiken ist Theorieresolution. Wir zeigen, Resolution und Kondensierung ohne zusätzliche Verfeinerungsstrategien ist ein Entscheidungsverfahren für die Basispfadlogik und bestimmte Erweiterungen. Wir präsentieren einen verbesserten Theorieunifikationsalgorithmus für die Logik *S4*, sowie ein geordnetes E-Resolutionskalkül mit Normalisierungsfunktionen. Wir zeigen außerdem, daß die optimierte funktionale Übersetzung auch auf Axiomenschemas anwendbar ist. Dies ermöglicht insbesondere die Einbettung von Modallogiken mit Zählquantoren in die Logik erster Stufe und deren Behandlung mittels eines Kalküls, das Resolution und Arithmetik verbindet.

Extended abstract

Propositional modal logics enjoy increasing popularity in various fields of computer science. Many propositional modal logics are decidable, but a serious problem in real applications is the lack of sophisticated modal theorem provers. Some theorem provers are available that are based on Gentzen or tableaux calculi, of which none have the level of sophistication that first-order theorem provers like OTTER or SPASS have. This thesis is concerned with providing resolution decision procedures for propositional modal logics. This goes hand in hand with providing an appropriate translation method from modal logic to first-order logic. The standard relational translation method is based on the possible worlds semantics. It transforms modal formulae to restricted quantifier expressions involving binary relations. The problem with this translation approach is the inherent non-determinism in relations, manifested in the non-termination of standard resolution procedures. Unrefined resolution is merely a semi-decision procedure for the relational translation. We will not be using the relational translation method. Instead, we will adopt an optimised form of the functional translation method, for which resolution is guaranteed to terminate, as we will show.

This thesis studies and formalises the *optimised functional translation* method for propositional modal logics and the treatment of inference in resolution procedures, in particular, theory resolution procedures. The optimised functional translation method is based on the *functional translation* method put forward by Ohlbach (1988a, 1991), Herzig (1989) and others. This approach follows the *functional semantics* of modal logic that defines accessibility between worlds by functions. Herzig noted that a certain optimisation is possible for propositional modal formulae. The optimisation allows for universal and existential quantifiers to be swapped arbitrarily. In the relational context this operation is not admissible. However, in *maximal* or *patched* functional models swapping quantifiers is satisfiability equivalence preserving. This property hinges on the generated frame property that embodies the fact that truth in a world of a modal formula does not depend on predecessor worlds.

The quantifier exchange operation is important for our decidability result, for it eliminates in the clausal forms all Skolem functions other than Skolem constants. Modal logics transform by the optimised functional translation to a lattice of clausal logics, called *path logics*. The weakest path logic is called *basic path logic* and is associated with the basic modal logics K and KD. It forms a fragment of monadic first-order logic with constant symbols and one binary function symbol, namely juxtaposition. Terms in basic path logic are strings of variables and constants that we view as *paths* in the underlying generated frame. The variables and constants are ordered as stipulated by the *prefix stability* property. It requires that different occurrences of one variable in any clause have the same prefix. We show that resolution and condensing without additional refinement strategies is a decision procedure for basic path logic and certain of its extensions. This result is important for three reasons. One, unrefined resolution and condensing provides a decision procedure for the translation

of many propositional modal logics, including K and arbitrary extensions with D, T and 4, and also *S5*. Two, any resolution procedure with condensing and *any* compatible refinement strategy is a decision procedure for the relevant modal and path logics. For practical purposes this is paramount, since any fair implementation of a resolution theorem prover can serve as a reasonable and efficient inference tool for doing basic modal reasoning. This is confirmed by a series of benchmarks done with OTTER, SPASS, and other special purpose theorem provers (Hustadt and Schmidt 1997a, 1997b, 1997c, Hustadt, Schmidt and Weidenbach 1998). Three, from a logical perspective, basic path logic appears to be the first solvable class (that is non-trivial) for which unrefined resolution and condensing solve the class.

The general calculus for path logics is theory resolution. We propose an E-unification algorithm for the path theory associated with *S4* that is more effecient than the algorithms available in the literature. Our algorithm combines the mutation rules for our identity law and associativity law, adapted from the general muation rules considered separately in Comon, Haberstrau and Jouannaud (1994) and Kirchner and Klay (1990). Mutation rules have the advantage that paramodulating into terms can be avoided. We introduce a general calculus of ordered E-resolution that is different from the calculus defined in Baumgartner (1992). Our calculus is based on the saturation approach of Bachmair and Ganzinger (1994) which admits a strong notion of redundancy that accommodates the simplification rules required for different path theories. In combination with the method of renaming, ordered E-resolution results in a considerable efficiency gain.

The optimised functional translation method has another advantage over the relational method. It applies not only to modal formulae, but we show the optimisation applies also to axiom schemas. A pleasant consequence is that some modal logics not determined by any elementary class of frames can be embedded in first-order logic. This extends the applicability for the resolution method (and other first-order theorem proving techniques) to essentially second-order modal logics, like K extended with McKinsey's schema. We make use of the new possibilities in a case study of accommodating graded modal logic in our first-order setting. Graded modal logic is important in many applications, especially in knowledge representation and computational linguistics, because it includes numerical quantifiers such as 'there are at least n' or 'there are more than n'. These operators are non-standard in the sense that their semantics is not in accordance with the semantics of $\Diamond$ or $\Box$. This complicates the translation to first-order logic. We introduce a new and more expressive modal logic with standard modalities that serves as an intermediary logic. The translation proceeds then in two steps: We translate graded modal logic into the new modal logic, followed by the optimised functional translation to first-order logic. The new modal logic includes a schema that is essentially second-order but which can be approximated in first-order logic by the optimised functional translation. We demonstrate how automated inference with counting operators can be realised in a calculus of resolution, paramodulation and basic arithmetic.

Ausführliche Zusammenfassung

In vielen Teilgebieten der Informatik spielen modale Aussagenlogiken eine tragende Rolle. Die gängigsten Modallogiken sind entscheidbar, was in Anwendungen enorm wichtig ist. Allerdings besteht ein Mangel an ausgereiften modalen Theorembeweisern. Der Entwicklungsgrad von Implementierungen von Tableau- oder Gentzenkalkülen für Modallogik ist eindeutig niedriger wie der von Beweisern für Logik erster Stufe wie zum Beispiel OTTER und SPASS. Diese Arbeit befaßt sich mit der Problemstellung, Resolutionsverfahren als Entscheidungsverfahren für modale Aussagenlogiken einzusetzen. Hierzu bedarf es einer geeigneten Übersetzung von Modallogiken in die Logik erster Stufe. Der bekannte semantikorientierte relationale Übersetzungsansatz basiert auf der Kripke-Semantik und transformiert modale Formeln zu Ausdrücken mit eingeschränkten Quantoren und binären Relationen. Ein Nachteil dieser Methode ist der Nichtdeterminismus inhärent in den Relationen und das offenkundige Problem der Nichtterminierung von Resolutionsverfahren. Unverfeinerte Resolutionsverfahren bieten für die relationale Übersetzung nur ein Semi-Entscheidungsverfahren. Wir werden in dieser Arbeit den relationalen Ansatz nicht weiter betrachten, sondern wenden uns einer Verfeinerung der funktionalen Übersetzung zu, die, wie wir zeigen werden, Terminierung von Resolutionsverfahren garantiert.

Die vorliegende Arbeit untersucht und formalisiert den *optimierten funktionalen Übersetzungsansatz* für modale Aussagenlogiken und dessen Behandlung durch Resolutionsverfahren, im allgemeinen durch Theorieresolutionsverfahren. Der optimierte funktionale Übersetzungsansatz basiert auf dem *funktionalen Übersetzungsansatz*, der von Ohlbach (1988a, 1991), Herzig (1989) und anderen für modale Prädikatenlogiken vorgestellt wurde. Dieser Ansatz bedient sich der *funktionalen Semantik* von Modallogik, in der die Erreichbarkeitsbeziehung zwischen Welten durch Funktionen modelliert wird. Herzig hat gezeigt, daß für Formeln in modalen Aussagenlogiken eine *Optimierung* der Übersetzungen möglich ist. Sie erlaubt beliebiges Vertauschen von Existenz- und Allquantoren. In der relationalen Semantik ist diese Operation nicht zulässig. In speziellen *maximalen* funktionalen Modellen ist das Vertauschen von Quantoren jedoch erfüllbarkeits- und unerfüllbarkeitserhaltend.

Die Quantorenvertauschungsoperation ist essenziell für unser Entscheidbarkeitsresultat. Damit lassen sich in der Klausalform der Übersetzung bis auf Skolemkonstanten komplexe Skolemterme ganz vermeiden. Der Verband der modalen Aussagenlogiken erzeugt mittels der optimierten funktionalen Übersetzung einen Verband von sogenannten *Pfadlogiken*. Dieses sind Klausellogiken, von denen die schwächste Pfadlogik, die *Basispfadlogik*, die den Logiken K und KD zugeordnet ist, ein monadisches Fragment der Logik erster Stufe mit Konstantensymbolen und einer zweistelligen Konkatenationsfunktion ist. Terme dieser Logik sind Worte, die aus Variablen und Konstantensymbolen gebildet werden und die *Pfade* in den zugrundeliegenden generierten Relationalstrukturen darstellen. Die Variablen und Konstanten sind einer speziellen Ordnung unterworfen, die durch die *Präfixstabilitätseigenschaft* festgelegt ist. Gemäß dieser Eigenschaft haben verschiedene Auftreten einer Variablen innerhalb einer Klausel einen eindeutigen Präfix. Wir zeigen, daß Resolution mit Kondensierung ohne zusätzliche Verfeinerungsstrategien ein Entscheidungsverfahren für die Basispfadlogik und bestimmte Erweiterungen ist. Dieses Resultat ist aus drei Gründen bemerkenswert. Erstens, ergeben sich daraus Resolutionsentscheidungsverfahren für die Übersetzung von mehreren Modallogiken, speziell K und beliebigen Erweiterungen mit den Schemata D, T und 4 und der Logik *S5*. Zweitens, ist jedes vollständige Resolutionsverfahren mit Kondensierung und *beliebigen* Verfeinerungsstrategien ein Entscheidungsverfahren für die relevanten Modal- und Pfadlogiken. Ein wichtiger praktischer Vorteil ist: Jeder resolutionsbasierter Beweiser, der das übliche Fairneßkriterium erfüllt, ist gleichzeitig ein akzeptabler und vergleichbar effizienter Beweiser für die Basismodallogiken. Dies bestätigt eine Reihe von Benchmarks mit OTTER, SPASS und anderen Spezialbeweisern (Hustadt und Schmidt 1997a, 1997b, 1997c, Hustadt, Schmidt und Weidenbach 1998). Drittens, vom logischen Gesichtspunkt ist bemerkenswert, daß die Basispfadlogik die erste lösbare (und nichttriviale) Klasse erster Stufe zu sein scheint, die im Resolutionskalkül mit Kondensierung ohne Verfeinerungen gelöst werden kann.

Das allgemeine Kalkül für Pfadlogiken ist Theorieresolution. Der von uns vorgestellte E-Unifikationsalgorithmus für *S4* is effizienter wie die herkömmlichen Algorithmen. Unser Algorithmus verbindet die Mutationsregeln für unser Identitäts- und Assoziativitätsgesetz aus den getrennt betrachteten allgemeinen Algorithmen von Comon et al. (1994) und Kirchner and Klay (1990). Mutationsregeln haben die schöne Eigenschaft, daß Unifikation innerhalb von Termen überflüssig wird. Wir führen ein geordnetes E-Resolutionsverfahren für Logik im Allgemeinen ein, welches im Unterschied zu Baumgartner (1992) auf dem Saturierungsansatz von Bachmair and Ganzinger (1994) basiert, der einen mächtigen Redundanzbegriff besitzt. Der Einsatz geordneter E-Resolution und der Technik der Umbenennung bringt einen Effizienzgewinn, den wir erörtern.

Der optimierte funktionale Übersetzungsansatz hat einen weiteren deutlichen Vorteil im Gegensatz zum relationalen Ansatz. Die Quantorenvertauschungsoperation ist nicht nur für modale aussagenlogische Formeln erlaubt, sondern wir zeigen, daß sie auch für Axiomschemata möglich ist. Bemerkenswert dabei ist vor allem, daß einige Modallogiken, die sich nicht durch eine Klasse von elementaren Relationalstrukturen charakterisieren lassen, so in Logik erster Stufe eingebettet werden können. Dies gilt zum Beispiel für die Erweiterung der Logik K mit dem McKinsey-Schema. Somit wird die Behandlung von solchen nichtelementaren Modallogiken durch Resolution (und anderen Techniken und Verfahren für Logik erster Stufe) ermöglicht. Die Tragweite dieser neuen Erkenntnis zeigt eine Fallstudie der Behandlung einer Modallogik mit Zählquantoren, im Englischen 'graded modal logic'. In Anwendungen, zum Beispiel in der Wissenrepräsentation und in der Computerlinguistik, sind Zählquantoren wichtig, denn sie eignen sich zum Modellieren von Ausdrücken wie zum Beispiel 'es gibt mindestens n' oder 'es gibt mehr wie n'. Die Semantik der Zählquantoren liegt quer zu der Semantik von $\diamond$ und $\Box$, was einigen Aufwand verursacht bei der Abbildung in die Logik erster Stufe. Unsere Übersetzung erfolgt in zwei Schritten: Wir führen eine neue mächtigere Modallogik ein, in die wir die Modallogik mit Zählquantoren zunächst abbilden, und die wir dann, wie gehabt, in Logik erster Stufe übersetzen. Eines der Axiome der neuen Zwischenlogik hat ähnlich wie das McKinsey-Schema keine äquivalente Formulierung mittels der relationalen Übersetzung. Wir zeigen jedoch, daß sich mittels der optimierten funktionalen Übersetzung eine in unserem Rahmen geeignete Formel erster Stufe angeben läßt. Somit kann die automatische Beweisführung für die Logik mit Zählquantoren in einem System realisiert werden, das Resolution, Paramodulation und elementare Arithmetik verknüpft.

Acknowledgements

First and foremost I thank Hans Jürgen Ohlbach for his continued support and encouragement. The cooperation with him greatly influenced and furthered my thinking about non-standard translation approaches and their value. He gave me several valuable ideas.

I am indebted to Harald Ganzinger and the Max-Planck-Institut for offering me employment and the pleasant working environment I was privileged to experience. A suggestion by Harald led to the considerable improvement of the proof of Theorem 6.5.1. An anonymous referee made a similar proposal.

My work has received tremendous stimulus from the cooperation with Ullrich Hustadt. He read much of this thesis very carefully and provided me with helpful suggestions. I benefited from frequent discussions with Andreas Nonnengart and Christoph Weidenbach who also read and scrutinised parts of my work. Collective thanks go to my colleagues in the institute, especially, Uwe Waldmann, Alexander Bockmayr, Sergei Vorobyov, Miroslava Tzakova, Luca Viganò, Jürgen Stuber and Georg Struth.

Finally, I wish to express my gratitude to Chris Brink who supervised my work for a Masters degree obtained from the University of Cape Town.

The PhD was supported financially by the Max-Planck Gesellschaft and two projects funded by the Deutsche Forschungsgemeinschaft, namely Tics (SFB 314) and TraLos.

Contents

Introduction

This thesis studies optimised functional translation methods of propositional modal logics into first-order logic and the behaviour of first-order resolution on these translations.

Propositional modal logics are increasingly being used in various fields of computer science, including knowledge representation, the field of logics of programs and computational linguistics. The popularity of modal logic can be attributed to its incredibly simple syntax, its natural semantics and decidability. The language of modal logic is a propositional language extended with modal quantifier operators $\Diamond_i$ and $\Box_i$. Semantically the fundamental concept is that of a relational structure defined by a set W of worlds and one or more binary (accessibility) relations on W. Just about anything in computer science is a relational structure: graphs are relational structures, orderings are relational structures, various notions of time are modelled by relational structures (linear or branching structures), labelled state transition systems for modelling programs and their correctness are relational structures, attribute value structures are relational structures, and relational structures underly semantic networks. It is no accident that modal logics have been invented in computer science (dynamic logic) and continue to be reinvented, for example in linguistics (feature logic) and in artificial intelligence (description logics). In many applications decidability is of immense importance, for example, in knowledge representation and in hardware or program specification and verification. Theoremhood in the basic modal logics K and its multi-modal version $K_{(m)}$ are decidable, and there are expressive extensions, like propositional dynamic logic, which remain decidable. Despite the popularity of propositional modal logic and their pleasant properties a serious problem in real applications is the lack of sophisticated modal theorem provers. Some theorem provers are available which are based on Gentzen or tableaux calculi. But, none have the level of sophistication that first-order theorem provers, like OTTER or SPASS, have.

Many attempts at using resolution theorem provers for modal reasoning have been frustrated by the fact that resolution is only a semi-decision procedure for first-order logic. Without any heuristics resolution continually produces more and more seemingly non-redundant clauses without terminating, not even for decidable fragments of first-order logic that correspond to decidable propositional modal logics. The standard translation of modal logics is based on the relational Kripke semantics, for which unrefined resolution is not a decision procedure. In the late eighties a number of researchers independently put forward an alternative translation of quantified modal logics to first-order logic, for which resolution theorem provers promised to be more efficient, and possibly terminating for propositional modal logics. This thesis studies this alternative translation approach, called the *functional translation* approach, and its *optimisation*, and exhibits what has often been claimed but never been substantiated: For many propositional modal logics unrefined resolution decides modal theoremhood and satisfiability.

Important for our decidability result is that we use an *optimised* version of the functional translation approach. It is obtained by swapping existential and universal quantifiers in a non-standard way. The optimisation is known to apply to propositional modal formulae. We show the optimised functional translation applies also to axiom schemas with a somewhat surprising spin-off that some essentially second-order modal logics, like K plus McKinsey's schema, become first-order under the optimised functional translation. This boosts the applicability of first-order theorem proving techniques for modal reasoning. We utilise this new-found ability to treat truly second-order modal logics in a case study of accommodating graded modal logic in a first-order context. Graded modal logic includes numerical quantifier operators which are important in many applications, like knowledge representation and computational linguistics. The counting operators are non-standard and require special treatment, as their semantics is not in accordance with the semantics of $\Diamond$ or $\Box$. Our solution is to translate graded modal logic first into a richer modal logic with standard modalities and then into first-order logic. The new modal logic includes a schema that is essentially second-order but has a first-order approximation by the optimised functional translation. We demonstrate how inference with counting operators can be formalised in a calculus of resolution, paramodulation and basic arithmetic.

By swapping quantifiers the optimised functional translation eliminates in the clausal forms all Skolem functions other than Skolem constants. It embeds modal logics in a hierarchy of clausal logics, called *path logics*. The *basic path logic* is associated with the basic modal logics K and $K_{(m)}$ as well as KD, and is a fragment of monadic first-order logic with function symbols. Basic path logic includes only nullary functions and one designated binary function, namely juxtaposition. Terms are lists (or strings) of variables and constants and encode accessibility as paths from the initial world. Terms, also called *paths*, satisfy the restriction that different occurrences of one variable in any clause have the same prefix. This property, called *prefix stability*, is essential for the decidability proof. We prove a general result that *any* complete resolution procedure with condensing is a decision procedure for the satisfiability problem of formulae in certain path logics, including basic path logic. The result is important for a number of reasons. It immediately follows that standard unrefined resolution is a decision procedure for many non-transitive uni- or multi-modal logics, including K and extensions with D and T as well as their multi-modal versions, and also *S5*. We also get decision procedures for the transitive modal logics *KD4* and *S4*, though these are not optimal. Exciting about the general decidability result is its practical value. Namely, many standard first-order theorem provers which are based on resolution are suitable for facilitating modal reasoning. Moreover, our result allows for *any* modifications and refinements, provided they are compatible with the basic calculus. This provides considerable freedom for experimentation toward attaining good performance, or for accommodating propositional modal logics also within specialised resolution procedures devised for extensions and combinations of modal logics. A refinement we study is ordered resolution, and more generally, ordered E-resolution. Practical experience with resolution theorem proving suggests our method competes very well with state-of-the-art tableaux or sequent-based theorem provers.

The thesis contains eight chapters. The following summarises their contents.

Chapter 1 offers by way of examples a preview of the fundamental ideas and results presented in this thesis.

The main theme of Chapters 2, 3 and 4 is the translation of propositional modal logics to first-order logic. Chapter 2 defines the essential concepts of propositional modal logic, its relational semantics and the relational translation approach, and gives a formal treatment of the functional semantics, its correspondence theory and the associated functional translation method. Chapter 3 describes the optimised functional translation approach. It investigates the transformation of second-order correspondence properties formulated in the relational language to possibly first-order properties formulated in the functional language. Chapter 4 presents the application of our techniques to graded modal logic. It demonstrates how inference with counting operators can be accommodated in a resolution calculus with paramodulation and the capability to perform basic arithmetical calculations.

Chapters 5, 6 and 7 are concerned with theory resolution and the problem of decidability by resolution. Chapter 5 introduces the hierarchy of path logics, the clausal logics to which modal logics translate by the optimised functional translation. It describes and studies the closure properties of path logics under unrefined resolution procedures with and without theory unification. Chapter 6 focuses on the decidability problem of path logics satisfying a term depth bound assumption. It proves that unrefined resolution is a decision procedure for path logics for which (i) a term depth bound exists, (ii) (theory) unification is decidable, and (iii) normalisation reduces terms to terms of basic path logic. Chapter 7 presents an ordered E-resolution calculus with normalisation and is concerned with a conversion to clausal form that uses the technique of renaming and produces clause sets of reduced complexity.

Chapter 8 is the conclusion and summarises the results obtained. To emphasise the practical value of the decision results the chapter also includes graphs of an empirical analysis of the performances of resolution theorem provers as compared to special purpose implementations for $K_{(m)}$.

The thesis concludes with a bibliography, an appendix, a list of figures, an index of notation, an index of schemas, rules and logics, and a subject index. Appendix A briefly describes the automated method to modal correspondence theory realised with a tool, called SCAN. Appendix B defines orderings on terms, atoms, literals and clauses, and gives the proof of ground completeness for ordered E-resolution (defined in Chapter 7).

Most parts of this thesis have been presented at various conferences and have been published or are due to be published in proceedings volumes or journals.

Chapter 3 (on the optimised functional approach) is based on joint work with Hans Jürgen Ohlbach and appears in the *Journal of Logic and Computation*. Chapter 4 (on graded modal logic) is based on joint work with Hans Jürgen Ohlbach and Ullrich Hustadt and is published in a collection on *Proof Theory of Modal Logic*. The presentation here corrects some misprints and includes some additional remarks. The notation has been adapted so as to be conform with the notation of this thesis. The performance analysis briefly mentioned in the Conclusion is joint work with Ullrich Hustadt which was presented at the *Fifteenth International Joint Conference on Artificial Intelligence*. The long version is to appear in the *Journal of Applied Non-Classical Logics*.

All other parts are entirely my own work. Chapter 2 (on the functional approach) is a formalisation from a semantic perspective of what is known, for the most, including some original results. The material of Chapter 5 (on path logic and theory resolution) can be viewed as combining and improving the ideas and methods of Ohlbach (1988a, 1991), Zamov (1989) and Auffray and Enjalbert (1992). The most important result, I feel, is the general decision result of Chapter 6 for a certain class of path logics. The first version of the proof is outlined in an extended abstract appearing in a volume of contributions to the *Advances in Modal Logic '96* workshop, and a detailed version is available as a research report. The new proof presented here is considerably simpler. The material of Chapter 7 (on ordered

E-resolution) has not been published before.

The following lists in chronological order the work which has been published or is due to be published.[1]

Functional translation and second-order frame properties of modal logics. Jointly with H. J. Ohlbach. *Research Report MPI-I-95-2-002.* Published in *Journal of Logic and Computation* **7**(5), 581–603 (1997).

Translating graded modalities into predicate logic. Jointly with H. J. Ohlbach and U. Hustadt. *Research Report MPI-I-95-2-008.* Published *in* H. Wansing (ed.), *Proof Theory of Modal Logic*, Vol. 2 of *Applied Logic Series*, Kluwer, pp. 253–291 (1996).

Symbolic arithmetical reasoning with qualified number restrictions. Jointly with H. J. Ohlbach and U. Hustadt. *In Proceedings of International Workshop on Description Logics'95*, Vol. 07.95 of *Rap.*, Univ. degli studia di Roma, pp. 89–95 (1995).

Resolution is a decision procedure for many propositional modal logics. Presented at the *Workshop on Advances in Modal Logic (AiML'96). Research Report MPI-I-97-2-002.* To appear in *in* M. Kracht, M. de Rijke, H. Wansing and M. Zakharyaschev (eds), *Advances in Modal Logic*, CSLI Publications, Stanford.

On evaluating decision procedures for modal logics. Jointly with U. Hustadt. *In* M. Pollack (ed.), *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI'97)*, Vol. 1, Morgan Kaufmann, pp. 202–207 (1997).

An empirical analysis of modal theorem provers. Jointly with U. Hustadt. To appear in the *Journal of Applied Non-Classical Logics.*

Relational grammars for knowledge representation. To appear in Böttner, M. (ed.), *Proceedings of the Workshop on Variable-Free Semantics*, Fachbereich Sprach- und Literaturwissenschaft, Univ. Osnabrück (1997).

E-unification for subsystems of S4. To appear in the *Proceedings of RTA'98*, *Lecture Notes in Computer Science*, Springer, Tsukuba, Japan (1998).

Optimised functional translation and resolution. Jointly with U. Hustadt and C. Weidenbach. To appear in the *Proceedings of TABLEAUX'98*, *Lecture Notes in Computer Science*, Springer, Oisterwijk, The Netherlands (1998).

[1] A list of my earlier work concerned with relation algebra, Peirce algebra and knowledge representation can be found at `http://www.mpi-sb.mpg.de/~schmidt/publications/`.

Chapter 1

Preview

This chapter introduces the optimised functional translation approach. It gives a preview of the most important ideas and methods used in this thesis. For illustration, because non-serial modal logics can be embedded in serial modal logics, we focus in this chapter on serial modal logics and frames with total accessibility relations.

1.1 The optimised functional translation

Just as the relational translation method of modal logic is based on its relational semantics, the functional translation method and the optimised form are based on its functional semantics. A *functional model* is a tuple (W, AF, ι) of a set of worlds, a set of so-called accessibility functions and a valuation mapping that determines the truth of atomic variables. The essential difference to the familiar notion of validity is that validity for modal expressions is defined in accordance with

$$(1.1)\qquad \begin{array}{lll} \mathcal{M}, x \models \Diamond\varphi & \text{iff} & \mathcal{M}, \alpha(x) \models \varphi \text{ for some function } \alpha \in AF \\ \mathcal{M}, x \models \Box\varphi & \text{iff} & \mathcal{M}, \alpha(x) \models \varphi \text{ for any function } \alpha \in AF. \end{array}$$

Figure 1.1 illustrates this definition.

The functional semantics can be viewed as arising from the relational semantics by a reformulation of relations as sets of functions. The key idea is that any binary relation R can be defined by a set of (partial or total) functions. In particular, any total relation can be defined by a set of total functions. For example, a set of total functions that defines the relation R depicted in Figure 1.2 is the set $AF_R = \{\alpha, \beta\}$ defined as illustrated in Figure 1.3(a). Any directed edge labelled by α linking two worlds x and y, say, indicates that the function α maps the world x to y, that is, $\alpha(x) = y$. This definition of R is not

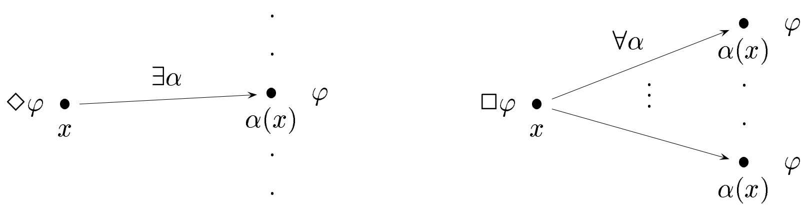


Figure 1.1: Validity in a functional model (W, AF, ι).

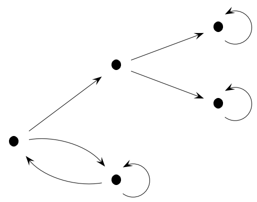
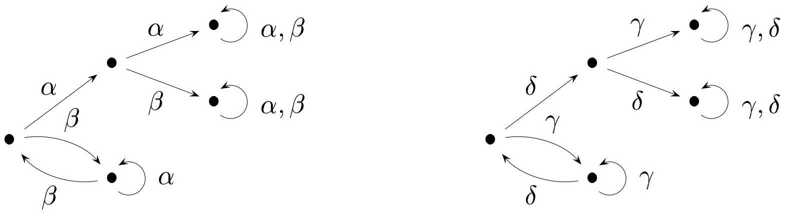


Figure 1.2: A sample binary relation R



(a) R defined by $\{\alpha, \beta\}$ (b) R defined by $\{\gamma, \delta\}$

Figure 1.3: Two functional encodings of R

unique. Another set of total functions that defines R is the set $\{\gamma, \delta\}$ with γ and δ defined as illustrated in Figure 1.3(b). The function γ is like α and δ is like β, except that γ maps the world at the root to its lower successor whereas α maps it to the upper successor, and δ maps it to the upper successor whereas β maps it to the lower one. A third definition of R is the set $AF_R = \{\alpha, \beta, \gamma, \delta\}$ including all four functions. The set of all total functions that define R is the largest defining set. In this example this set contains eight functions (its cardinality is determined by the amount of branching in R).

There are many *different* functional frames which represent the *same* relational frame. One such functional frame is sufficient for proving the existence of a model. This means, we have some freedom in choosing a functional frame which is suited best for our purposes. The special frame we use is the frame in which the defining set of functions is the set of *all* total functions that define R. This frame is referred to as the *maximal functional frame*. We will prove any modal logic complete in the relational semantics is also complete with respect to the class of all functional frames, and more important, it is also complete with respect to the class of all maximal functional frames. In maximal frames enough functions are available, which can be exploited in reducing variable dependencies in (i) the functional correspondence properties and in (ii) the optimised functional translation.

The *functional correspondence* property of the axiom schema 4 in a serial context is the formula

$$\forall x\ \forall \alpha\beta\exists\gamma\ \beta(\alpha(x)) = \gamma(x)$$

which expresses 'local composability': Viewed from inside any world x, the function γ is the composition of α and β. In the maximal frames, as all functions defining a given total transitive relation are available, in particular, also the composition of any pair of functions, 'global composability'

$$\forall \alpha\beta\exists\gamma\ \forall x\ \beta(\alpha(x)) = \gamma(x) \tag{1.2}$$

is true. It is open to which extent this principle generalises. We know transitivity and reflexivity are accommodated by their global forms, visualised in Figure 1.4. Global identity

4 transitivity

T reflexivity

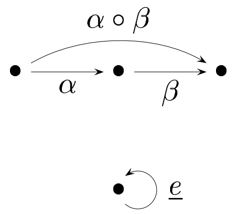


Figure 1.4: Illustration of functional correspondence properties.

is given by

$$\exists\alpha\ \forall x\ \alpha(x) = x, \tag{1.3}$$

and defines α to be the identity function (denoted by $\underline{e}$). In these instances the elimination of the dependency on the world variable preserves both satisfiability and unsatisfiability because the respective classes of maximal frames are closed under functional composition and identity.

The following movement of quantifiers over functions is also possible:

$$\exists\alpha\forall\beta\ \psi \quad \text{becomes} \quad \forall\beta\exists\alpha\ \psi, \tag{1.4}$$

which is based on the equivalence

$$\exists\alpha\forall\beta\ \psi \leftrightarrow \forall\beta\exists\alpha\ \psi$$

true in any maximal frame. We give an informal argument, why this is this case. Consider the model of Figure 1.5. The formula $\Box\Diamond p$ is true in the root x because both its successor worlds y_1 and y_2 each have a successor in which p is true, namely z_1 and z_3. In the associated maximal functional model the defining set of accessibility functions is a set $AF_R = \{\alpha_1, \dots, \alpha_8\}$ with cardinality eight. Suppose in this model the following is not the case.

$$\forall\alpha\exists\beta\ \text{`}p \text{ is true in } \beta(\alpha(x))\text{'} \rightarrow \exists\beta\forall\alpha\ \text{`}p \text{ is true in } \beta(\alpha(x))\text{'}. \tag{1.5}$$

Suppose the antecedent is true, but no β in AF_R is such that for every function α, p is true in the world $\beta(\alpha(x))$. Assume $\alpha_1, \dots, \alpha_4$ map y_1 to z_1 and $\alpha_5, \dots, \alpha_8$ map y_1 to z_2, and $\alpha_5, \dots, \alpha_8$ map y_2 to z_3 and $\alpha_1, \dots, \alpha_4$ map y_2 to z_4. Because AF_R is the set of all total functions encoding R there must be a function among the $\alpha_5, \alpha_6, \alpha_7, \alpha_8$ which is like α_1 for any world except for the world y_2. In particular, this function α_5, say, assigns y_2 to the same world as α_1 does. This contradicts the assumption that α_1 maps y_2 to z_4. Hence, β can be taken to be the function α_1 which proves (1.5).

The *optimised functional translation* of a modal formula φ is obtained by a sequence of two transformations: (i) the *functional translation* mapping which mimics the definition of truth in any functional model, and (ii) the so-called *quantifier exchange* operator Υ, which swaps quantifiers according the rule (1.4). We will illustrate the conversion to first-order logic by way of an example. In order to get first-order formulations the functional translation simulates terms of the form $\beta(\alpha(x))$ by $[[x, \alpha], \beta]$, abbreviated $[x\alpha\beta]$, using the functional application operation $[\cdot, \cdot]$. According to (1.1), the functional translation of a diamond formula and its dual is specified by

$$\pi_f(\Diamond\varphi, s) = \exists\alpha\ \pi_f(\varphi, [s\alpha]) \quad \text{and} \quad \pi_f(\Box\varphi, s) = \forall\alpha\ \pi_f(\varphi, [s\alpha]),$$

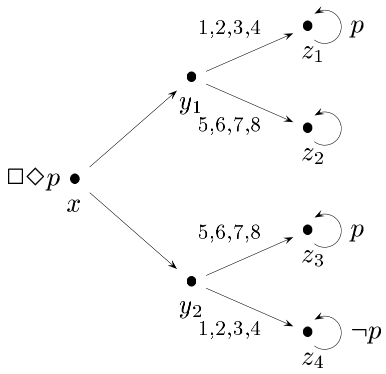


Figure 1.5: A relational model of R

where s has one of two forms, either x or $[x\beta_1 \ldots \beta_m]$, and α is a variable that does not occur in s. The following formula is a theorem in the serial modal logic KD.

$$\text{(1.6)} \qquad \begin{array}{l} \rho_1 = \Box(\Diamond\Box\neg p \vee \Diamond\Diamond p) \\ \qquad\ \ \alpha\ \beta\ \gamma \qquad\ \ \delta\ \epsilon \end{array}$$

Its functional translation is

$$\Pi_f(\rho_1) = \forall x\ \pi_f(\rho_1, x) = \forall x\ \forall\alpha\ (\exists\beta\forall\gamma\ \neg P[x\alpha\beta\gamma] \vee \exists\delta\exists\epsilon\ P[x\alpha\delta\epsilon]).$$

The propositional symbol p in ρ_1 translates to a monadic predicate P. The boxes translate to universal functional quantifiers and the diamonds to existential functional quantifiers, in such a way that each occurrence of a modal operator is uniquely associated with a functional variable, as indicated in (1.6). The quantifier exchange operator moves all existential quantifiers inward and we get:

$$\Upsilon\Pi_f(\rho_1) = \forall x\ \forall\alpha\ (\forall\gamma\exists\beta\ \neg P[x\alpha\beta\gamma] \vee \exists\delta\exists\epsilon\ P[x\alpha\delta\epsilon]).$$

The order of the quantifiers in the prefix of the first part of the disjunction has been reversed. Negation produces

$$\neg\Upsilon\Pi_f(\rho_1) = \exists x\ \exists\alpha\ (\exists\gamma\forall\beta\ P[x\alpha\beta\gamma] \wedge \forall\delta\forall\epsilon\ \neg P[x\alpha\delta\epsilon]),$$

in which all existential quantifiers precede all universal quantifiers. Its clausal form is:

$$\begin{array}{ll} 1. & P[\underline{x}\underline{\alpha}\beta\underline{\gamma}] \\ 2. & \neg P[\underline{x}\underline{\alpha}\delta\epsilon] \end{array} \quad \text{which simplifies to} \quad \begin{array}{ll} 1. & P[\underline{\alpha}\beta\underline{\gamma}] \\ 2. & \neg P[\underline{\alpha}\delta\epsilon] \end{array},$$

because the constant $\underline{x}$ occurs in every term and carries no information. Formally, $\underline{x}$ is replaced by the 'empty list' []. Underlined symbols denote constants and non-underlined symbols denote variables. One resolution step is possible and produces the empty clause. This proves the formula $\neg\Upsilon\Pi_f(\rho_1)$ is unsatisfiable, and consequently, ρ_1 is a theorem of KD. The optimisation obtained with Υ has the pleasant and important property that, the Skolemised clausal form of $\neg\Upsilon\Pi_f(\varphi)$ for any φ does not contain any Skolem terms other than constants.

By completeness with respect to maximal frames it follows that for any complete modal logic $K\Sigma$, φ is provable in $K\Sigma$ iff $\Upsilon\Pi_f(\Sigma) \wedge \neg\Upsilon\Pi_f(\varphi)$ is unsatisfiable in second-order logic. When $\Upsilon\Pi_f(\Sigma)$ is equivalent to a first-order formula then

$$\varphi \text{ is provable in } K\Sigma \quad \text{iff} \quad \Upsilon\Pi_f(\Sigma) \wedge \neg\Upsilon\Pi_f(\varphi) \text{ is first-order unsatisfiable.}$$

This kind of statement holds for any translation approach that is sound and complete, in particular, for the relational translation Π_r: For any complete and first-order definable logic $K\Sigma$, provability reduces to testing unsatisfiability of $\Pi_r(\Sigma) \wedge \neg\Pi_r(\varphi)$ in first-order logic. Since completeness does not imply first-order definability the condition cannot be relaxed.

One important advantage we claim for the optimised functional translation is that $\Upsilon\Pi_f(\Sigma)$ is first-order for a wider class of logics than $\Pi_r(\Sigma)$ is. In general, modal axiom schemas translate to second-order formulae with quantifiers over the unary predicate symbols. Few axiom schemas are first-order definable. This limits the applicability of first-order deduction to modal logics that are complete with respect to a class of frames restricted by a set of first-order properties. There are modal logics, like K plus McKinsey's schema M, which are complete with respect to some class of frames, but which are essentially second-order, meaning provably no first-order relational characterisations exist. The second-order quantifier elimination method of Gabbay and Ohlbach (1992) implemented in SCAN pinpoints the problem. SCAN automatically reduces the input for M to a set of clauses free of unary predicates but it is impossible to reconstruct the existential quantifiers for the Skolem functions f and g in a linear fashion, since f depends on y_1 and g depends on z_1 and both $f(y_1)$ and $g(z_1)$ occur in the same clause:

$$\begin{aligned}
&\neg R(\underline{x}, y_1) \vee R(y_1, f(y_1)) \\
&\neg R(\underline{x}, z_1) \vee R(z_1, g(z_1)) \\
&\neg R(\underline{x}, y_1) \vee \neg R(\underline{x}, z_1) \vee f(y_1) \neq g(z_1).
\end{aligned}$$

Swapping existential and universal quantifiers makes the hampering variable dependencies disappear, and a remarkable property of the functional translation and the optimisation operator Υ is that this is exactly what happens for M. M is first-order definable in maximal functional frames. Via the optimised functional translation, first-order deduction is now applicable to a larger class of modal logics.

The structure of any modal formula determines a characteristic ordering of the variables in the terms of the functional and also optimised functional translation. This ordering is captured in the notion of *prefix stability*, which says that any variable in $\Pi_f(\varphi)$, $\Upsilon\Pi_f(\varphi)$ and their negations has a unique prefix, since any modal operator is uniquely associated with a variable. For example, in (1.6) the unique prefix of α in any term of $\Pi_f(\rho_1)$, $\Upsilon\Pi_f(\rho_1)$ and $\neg\Upsilon\Pi_f(\rho_1)$ is x, the unique prefix of β is $[x\alpha]$, etcetera. Prefix stability is a fundamental concept that is exploited in many results of this thesis, including the proof that quantifiers may be swapped, decidability of theory unification and decidability of resolution. The prefix stability property reflects that frames can be assumed to be generated frames which embodies the pleasant property of propositional modal logic that truth of a formula in a world does not depend on predecessor worlds. We advance that this is another advantage of the functional encodings over the relational encoding. The functional translations admit a new perspective of modal logic that is *path* oriented as opposed to world oriented. In functional models for $\Diamond\varphi$ to be true at a world x we require the existence a *one step path* (or function) that leads to a world at which φ is true. Evaluating the truth of a formula

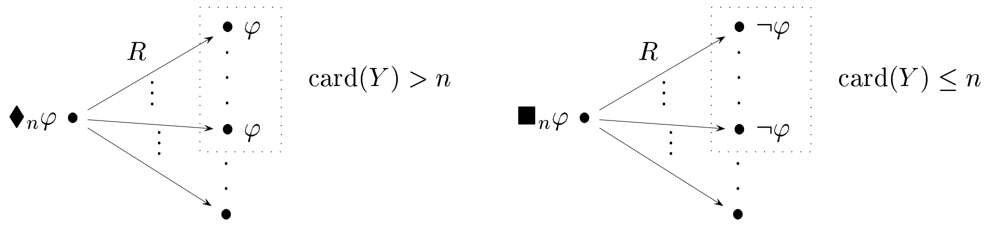


Figure 1.6: The semantics of $\blacklozenge_n\varphi$ and $\blacksquare_n\varphi$.

of modal depth greater than one requires following *paths* (determined by compositions of functions) from the initial world. Thus, we will view terms of the form $[x\alpha_1 \ldots \alpha_n]$ or $[\alpha_1 \ldots \alpha_n]$ to be paths from worlds x or $[]$, respectively.

1.2 A case analysis: Translating graded modal logic

In basic modal logic it is not possible to express numerical information, like 'a city is a place with *more than* 100 000 inhabitants'. There are extensions of $K_{(m)}$ with additional numerical quantifier operators, called *graded modalities*. In this context

$$\mathit{city} = \mathit{place} \ \wedge \ \blacklozenge_{100\,000}\mathit{people} \tag{1.7}$$

is a suitable definition. Graded modalities are (uni-modal or multi-modal) modal operators indexed with cardinals which fix the number of worlds in which a formula is true. A formula $\blacklozenge_n\varphi$ is true in a world iff there are more than n worlds accessible by a given R in which φ is also true. The dual formula $\blacksquare_n\varphi$, given by $\neg\blacklozenge_n\neg\varphi$, is then true in a world iff there are at most n accessible worlds in which $\neg\varphi$ is true. More formally, the semantics is defined in terms of *one* accessibility relation R by

$$\begin{aligned}
\mathcal{M}, x &\models \blacklozenge_n\varphi \quad \text{iff} \quad \mathrm{card}(\{y \mid R(x,y) \text{ and } \mathcal{M}, y \models \varphi\}) > n \\
\mathcal{M}, x &\models \blacksquare_n\varphi \quad \text{iff} \quad \mathrm{card}(\{y \mid R(x,y) \text{ and } \mathcal{M}, y \models \neg\varphi\}) \leq n,
\end{aligned}$$

as is illustrated in Figure 1.6. (For any set A, card(A) denotes the cardinality of A.) In the multi-modal context graded modalities have the form $\blacklozenge_{i,n}$ or $\blacksquare_{i,n}$ and are defined by a family of relations. We will restrict our attention to graded modalities in a uni-modal context as defined inside the graded modal logic $\overline{K}$.

The semantics is very natural and intuitive, but it has one disadvantage. Most inference systems based on this semantics, in particular, tableaux systems, deal with the $\blacklozenge_n$-operators by generating a corresponding number of worlds explicitly. For example, the formula $\blacklozenge_{100\,000}\mathit{people}$ triggers the generation of 100 001 constant symbols as representatives for the individuals denoting *people*. Except for counting these constant symbols and for comparing the length of lists, known tableaux systems do not provide for arithmetical computation. In particular, reasoning with symbolic arithmetic terms is impossible. For example, in tableaux systems the formula $\blacklozenge_{n+1}p \rightarrow \blacklozenge_n p$ which is true for all n can only be verified for concrete values of n, but in general it cannot be verified for arbitrary values of n.

A direct translation of formulae with graded modalities into predicate logic requires the axiomatisation of finite domains. This is feasible only for small cardinalities. We may translate sentence (1.7) as follows:

$$
\begin{aligned}
\forall x\ (\mathit{city}(x) \leftrightarrow (\mathit{place}(x) \wedge \exists y_1 \ldots y_{100\,001}\quad (& y_1 \neq y_2 \wedge y_1 \neq y_3 \wedge \ldots \wedge y_1 \neq y_{100\,001} \wedge \\
& y_2 \neq y_3 \wedge \ldots \wedge y_2 \neq y_{100\,001} \wedge \\
& \vdots \\
& y_{100\,000} \neq y_{100\,001} \wedge \\
& \mathit{inhabited\text{-}by}(x, y_1) \wedge \ldots \wedge \mathit{inhabited\text{-}by}(x, y_{100\,001}) \wedge \\
& \mathit{people}(y_1) \wedge \ldots \wedge \mathit{people}(y_{100\,001})))).
\end{aligned}
$$

The translation of $\blacklozenge_n$-expressions requires $(n+1)n/2$ in-equations. Even for small n this is more than current theorem provers can cope with. One immediate alternative is introducing set variables and a cardinality function, with which (1.7) can be formulated by:

$$
\begin{aligned}
\forall x\ (\mathit{city}(x) \leftrightarrow \mathit{place}(x) \wedge \exists Y\ (&\mathrm{card}(Y) > 100\,000 \wedge \\
&\forall y\ (y \in Y \rightarrow (\mathit{inhabited\text{-}by}(x, y) \wedge \mathit{people}(y))))).
\end{aligned}
$$

This is not a feasible alternative either, for the axiomatisation of the cardinality function then requires the above $(n+1)n/2$ in-equations, and this for every n:

$$
\begin{aligned}
\forall Y\ (\mathrm{card}(Y) > n \leftrightarrow \exists y_1 \ldots y_{n+1}\ \ (& y_1 \in Y \wedge \ldots \wedge y_{n+1} \in Y \wedge \\
& y_1 \neq y_2 \wedge y_1 \neq y_3 \wedge \ldots \wedge y_1 \neq y_{n+1} \wedge \\
& y_2 \neq y_3 \wedge \ldots \wedge y_2 \neq y_{n+1} \wedge \\
& \vdots \\
& y_n \neq y_{n+1})).
\end{aligned}
$$

We present a two step translation of graded modal logics into (sorted) first-order logic. In the first step, we transform graded modal logics into a multi-modal logic with the standard semantics. In the second step we perform the optimised functional translation. The first transformation accommodates modal logics of graded modalities, like $\overline{K}$, in a new multi-modal logic, called $\overline{K}_E$, with two kinds of modalities:

(i) $\Diamond_n$ and $\Box_n$ are characterised by a relational frame of infinitely but countably many different relations R_n (with n a non-negative integer), and

(ii) $\Diamond$ and $\Box$ are characterised by a designated relation E.

More specifically, we translate formulae of the form $\blacklozenge_n\varphi$ into $\Diamond_n\Box\varphi$ and the intuitive idea underlying this translation is as in Figure 1.7. If φ is true in a set Y of worlds with more than n elements then we introduce an accessibility relation R_n that connects the actual world and a world x_Y which we can think of as being a representative for the set Y.[1] This defines the $\Diamond_n$-operator. $\Box\varphi$ and its associated accessibility relation E expresses that φ is true in all the worlds of the set Y. E connects the world x_Y with all the worlds in Y and can be thought of as the membership relation. Thus, $\Diamond_n\Box\varphi$ encodes 'there is a set with more than n elements (encoded by $\Diamond_n$) and φ is true for all the elements of this set (encoded by $\Box$)'. Our first problem now is to find a sound and complete axiomatisation of the modalities $\Diamond_n$, $\Box_n$, $\Diamond$ and $\Box$ that accommodates the graded modalities $\blacklozenge_n$ and $\blacksquare_n$. It turns out that the new logic is richer and has some non-standard models which do not reflect our intuition, but the logic encompasses the graded modal logic. We show, a formula φ is a theorem of a graded modal logic iff the translation of φ is a theorem in the new logic. This translation from $\overline{K}$ to $\overline{K}_E$ is only an intermediary step in the transformation to first-order logic.

[1] We do not introduce a sort Y.

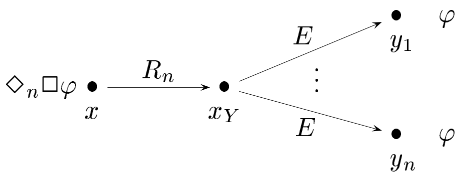


Figure 1.7: The intended semantics of $\overline{K}_E$

In the second step, we translate the multi-modal logic $\overline{K}_E$ into first-order logic using the optimised functional translation. $\overline{K}_E$ includes an axiom that is not first-order definable in terms of the R_n. But like McKinsey's axiom, this axiom reduces to a first-order formula when using the optimised functional translation.

This sequence of translations of a system of numerical modalities first into another multi-modal logic and then into a sorted first-order logic yields an axiomatisation in terms of a set of clauses. This axiomatisation can be viewed as defining properties of finite sets. Instead of counting symbols our proposed resolution calculus will also do arithmetical calculations.

1.3 Path logics and resolution

The optimised functional translation embeds the lattice of complete and first-order definable modal logics (by the optimised translation) into a corresponding lattice of first-order logics, which we call *path logics*. The name is motivated by our new perspective of the modal semantics. Path logics are clausal logics. The weakest of them is called *basic path logic*, because it is associated with the modal logics K and KD which are regarded as being basic in the functional context. We show that any non-serial modal logic can be embedded in a serial modal logic and a special case is that K can be embedded in KD adjoined with a new propositional variable. Clauses of basic path logic have the form

$$P[\underline{\alpha}\beta\underline{\gamma}] \vee \neg Q[\underline{\alpha}\delta\epsilon]$$

and are built from constant symbols, like $\underline{\alpha}$ and $\underline{\gamma}$, variables, like β and δ, a special constant symbol $[]$, a left associative operation $[\cdot,\cdot]$ and unary predicate symbols, like P and Q, as well as $\vee$ and $\neg$. The only Skolem terms in basic path clauses are constants. Terms, like $[\underline{\alpha}\beta\underline{\gamma}]$ and $[\underline{\alpha}\delta\epsilon]$, which we call paths, are required to satisfy prefix stability for variables. This is an important language restriction that avoids chaining of variables as in the clause

$$P[\alpha\beta] \vee Q[\beta\gamma] \vee R[\alpha\gamma].$$

Neither β nor γ have a unique prefix. Chaining of variables is known to cause all sorts of difficulties in connection with theory unification and resolution. Prefix stability is an invariance property with respect to the fundamental operations of resolution procedures, in particular, forming resolvents, factoring and condensing. Extensions of K and KD with different sets Σ of schemas give rise to different path logics determined by the corresponding theories $\Upsilon\Pi_f(\Sigma)$. The theories associated with T and 4 involve the formulae (1.3) and (1.2). Their respective Skolemised formulations are:

$$\begin{aligned} &\textit{right identity}: \quad [x\underline{e}] = x \\ &\textit{associativity}: \quad [x(\alpha \circ \beta)] = [x\alpha\beta]. \end{aligned}$$

Inference for basic path logics is facilitated by resolution with syntactic unification as we demonstrated in a sample derivation for ρ_1, in (1.6) above. Inference for non-basic path logics is facilitated by theory resolution. Theory resolution combines resolution with normalisation and a more sophisticated unification algorithm for solving equations modulo a given equational theory (or more generally, for testing unsatisfiability with respect to a consistent set of clauses that defines the given theory). This thesis focuses exclusively on equational theories for serial modal logics, concentrating on the theories defined by the equations for T and 4, above. These determine rewrite relations, from left to right as presented above, that produce normal forms which will be computed by recursive normalisation functions N_T and N_4.

To illustrate the envisaged resolution procedures we consider two examples. The following two clauses

$$\begin{array}{ll} 1. & P[\alpha] \\ 2. & \neg P[\alpha'\beta] \vee Q[\alpha'\beta] \end{array}$$

have no resolvents modulo the empty theory, because $P[\alpha]$ and $P[\alpha'\beta]$ have different lengths. Remember $[\alpha]$ is our short hand notation for $\alpha([])$ and $[\alpha'\beta]$ for $\beta(\alpha'([]))$. However, modulo right identity $P[\alpha]$ and $P[\alpha'\beta]$ have a most general unifier, namely $\{\alpha' \mapsto \alpha, \beta \mapsto \underline{e}\}$. The resolvent is:

$$3. \quad Q[\alpha\underline{e}] \qquad\qquad [1.1, 2.1, \text{right identity}]$$

$Q[\alpha\underline{e}]$ is logically equivalent to the simpler clause $Q[\alpha]$. This kind of simplification will be achieved with normalisation functions applied eagerly. Here, the normalisation function N_T will replace clause 3. by:

$$3'. \quad Q[\alpha]. \qquad\qquad [3, N_T]$$

Unification modulo the associativity law is realised by an algorithm akin to the first unification algorithm for general associativity proposed by Plotkin (1972). It introduces new variables and unfortunately normalisation does not prevent term growth, which this example illustrates:

$$\begin{array}{ll} 1. & P[\alpha\gamma] \vee Q[\alpha\gamma] \\ 2. & \neg P[\alpha'\underline{\beta}] \vee R[\alpha'\underline{\beta}]. \end{array}$$

A most general unifier of $P[\alpha\gamma]$ and $P[\alpha'\underline{\beta}]$ is $\{\gamma \mapsto \gamma' \circ \underline{\beta},\ \ \alpha' \mapsto \alpha \circ \gamma'\}$ with γ' being a new variable. Thus, a resolvent of 1. and 2. is:

$$3. \quad Q[\alpha(\gamma' \circ \underline{\beta})] \vee R[\alpha(\gamma' \circ \underline{\beta})]. \qquad\qquad [1.1, 2.1, \text{associativity}]$$

The normalisation function N_4 for associativity reorganises terms according to the rewrite rule $[x(\alpha \circ \beta)] \Rightarrow [x\alpha\beta]$ replacing clause 3. by the following.

$$3'. \quad Q[\alpha\gamma'\underline{\beta}] \vee R[\alpha\gamma'\underline{\beta}]. \qquad\qquad [3, N_4]$$

Unification for paths under right identity is decidable. Unification under associativity is also decidable, since paths are linear terms. We will present an improved unification algorithm for right identity and associativity that employs mutation rules, which have the advantage over lazy paramodulation rules that they apply to the top symbols of any pair of terms, making paramodulating into terms superfluous.

1.4 Decidability by unrefined resolution

We know that any complete resolution procedure $\mathcal{R}$ is a semi-decision procedure for first-order logic. Given any unsatisfiable set S of clauses the procedure is guaranteed to produce a contradiction in the form of the empty clause. But, the computation will not necessarily halt if S is a satisfiable set. When $\mathcal{R}$ halts without producing the empty clause then the input set S is satisfiable. Deduction also stops when no new clauses can be produced which are not variants of clauses already present.

We consider in an example how resolution continually produces more and more clauses with increasing complexity, for the relational translation. Proving the satisfiability of $\rho_2 = \Box(p \to \Diamond p)$ amounts to proving the satisfiability of the following set of clauses:

1. $\neg R(\underline{x}, y) \vee \neg P(y) \vee R(y, f(y))$
2. $\neg R(\underline{x}, y) \vee \neg P(y) \vee P(f(y))$.

($\underline{x}$ denotes a constant.) The clauses have two resolvents.

3. $\neg R(\underline{x}, \underline{x}) \vee \neg P(\underline{x}) \vee \neg P(f(\underline{x})) \vee \neg P(f^2(\underline{x}))$ [1.3, 2.1]
4. $\neg R(\underline{x}, f(y)) \vee R(f(y), f^2(y)) \vee \neg R(\underline{x}, y) \vee \neg P(y)$ [1.2, 2.3]

Clause 4. resolves with clause 2. and yields:

5. $\neg R(\underline{x}, f^2(y)) \vee R(f^2(y), f^3(y)) \vee \neg R(\underline{x}, f(y)) \vee \neg R(\underline{x}, y) \vee \neg P(y)$ [2.3, 4.4]

This clause also resolves with 2., and again, the resulting resolvent resolves with 2., and so forth. Repeatedly resolving the new resolvents with 2. yields increasingly longer and more complex clauses. None of the clauses is redundant and can be deleted. In the limit, our sample input set S has infinitely many non-variant resolvents. This shows standard unrefined resolution does not terminate for relational translations of modal formulae.[2] (I leave it to the reader to verify that the computation terminates for the optimised functional translation of ρ_2.)

The example demonstrates, one reason for an unbounded number of non-redundant resolvents being produced is:

(a) The level of nesting of the terms increases indefinitely (in the example, $f(y)$, $f^2(y)$, $f^3(y)$, and so on).

Another reason is:

(b) The number of literals in the resolvents increases indefinitely.

The idea of our decision proof for the optimised functional translation is that of Joyner Jr. (1976), which has become standard. The aim is to show that any resolution procedure combined with condensing, given by $\mathcal{R}_{\text{COND}}$, avoids the situations (a) and (b). This is done by considering any class $\mathbf{S}$ of path clauses generated by saturation upon any finite input set S, and exhibiting for the class,

[2] There are refinements of resolution that guarantee termination for the relational translation, for which refer to Fermüller, Leitsch, Tammet and Zamov (1993) and Hustadt (1997).

(A) the existence of a term depth bound, as well as

(B) the existence of a bound on the size (cardinality) of any clause

It is important to note that (B) is tied with the maximal number of variables which can occur in any clause. Given that there is a finite supply of predicate symbols and constant symbols (those of S), if a bound exists for the number of variables of any clause then the number of literals in any clause is bounded.

It turns out, no refinements are necessary as unrestricted resolution and condensing decides satisfiability for basic path logic as well as certain path logics for which (A) holds. This follows from a very general decision result:

> Assuming that a term depth bound exists, there is also a bound for the size of any clause in the class **S** of all non-variant basic path clauses built from a finite supply of constant and predicate symbols.

As can be seen in the derivation for ρ_1 above, syntactic unification does not expand terms, as bindings either rename variables or instantiate variables with constants, and the term depth bound assumption is true. Consequently, any complete resolution procedure that includes condensing is a decision procedure for basic path logic and the associated modal logics, namely K and KD, as well as *S5*. The result has more general consequences. The result applies to any theory resolution procedures (unrefined or refined) and to path logics for which (i) a term depth bound exists, (ii) (theory) unification is decidable, and (iii) normalisation eliminates non-constant functional terms introduced by unification. These conditions hold for resolution modulo right identity and associativity. For associativity, (i) can be met by an artificial term depth bound extracted from the literature. Thus, decidability follows also for KT, *KD4* and *S4*.

Suppose a term depth bound exists, that is, (A) holds. We aim at proving (B), namely, exhibiting a bound for the number of literals and/or variables in any clause of **S**. The following example illustrates that variable numbers may increase during resolution.

1. $P[\alpha] \lor Q[\alpha\beta]$
2. $\neg P[\alpha'] \lor R[\alpha'\gamma]$
3. $Q[\alpha\beta] \lor R[\alpha\gamma]$ [1.1, 2.1, empty theory]

Some superfluous variables and literals can be eliminated by condensing, though not in this example. A clause is condensed if there is no substitution σ such that $C\sigma \subsetneqq C$, that is, if it does not subsume a proper factor of itself. Condensing reduces

$$\begin{array}{llll} P[\alpha] \lor P[\underline{\beta}] & \text{to} & P[\underline{\beta}] & \text{by the substitution } \{\alpha \mapsto \underline{\beta}\}\text{, and} \\ P[\alpha] \lor P[\gamma] & \text{to} & P[\gamma] & \text{by } \{\alpha \mapsto \gamma\}. \end{array}$$

The pairs are logically equivalent (since, for instance, $P[\underline{\beta}]$ logically implies $P[\alpha] \lor P[\underline{\beta}]$ and $P[\alpha] \lor P[\underline{\beta}]$ subsumes and implies $P[\underline{\beta}]$).

The following example gives some indication of the potential size of clauses in **S**.

$$\begin{array}{l} P[\alpha\,\beta_1] \lor P[\alpha\,\beta_1\,\gamma_1] \lor Q[\alpha\,\beta_1\,\gamma_2] \\ \lor Q[\alpha\,\beta_2] \lor P[\alpha\,\beta_2\,\gamma_3] \lor Q[\alpha\,\beta_2\,\gamma_4] \lor P[\alpha\,\beta_3\,\gamma_5] \lor Q[\alpha\,\beta_3\,\gamma_5] \end{array} \tag{1.8}$$

It belongs to the class of clauses with maximal path length $m = 3$ and the number of predicate symbols plus constant symbols is $k = 2$. The clause has 8 literals and $1 + 3 + 5$

$$
\begin{array}{|l|}
\hline
\alpha\,\beta_1\;\underline{b}\;\underline{p}\\
\alpha\,\beta_1\;\gamma_1\;\underline{p}\\
\alpha\,\beta_1\;\gamma_2\;\underline{q}\\
\alpha\,\beta_2\;\underline{b}\;\underline{q}\\
\alpha\,\beta_2\;\gamma_3\;\underline{p}\\
\alpha\,\beta_2\;\gamma_4\;\underline{q}\\
\alpha\,\beta_3\;\gamma_5\;\underline{p}\\
\alpha\,\beta_3\;\gamma_5\;\underline{q}\\
\hline
\end{array}
\qquad
\begin{array}{|l|l|}
\hline
\alpha & \beta_1\;\underline{b}\;\underline{p}\\
\alpha & \beta_1\;\gamma_1\;\underline{p}\\
\alpha & \beta_1\;\gamma_2\;\underline{q}\\
\alpha & \beta_2\;\underline{b}\;\underline{q}\\
\alpha & \beta_2\;\gamma_3\;\underline{p}\\
\alpha & \beta_2\;\gamma_4\;\underline{q}\\
\alpha & \beta_3\;\gamma_5\;\underline{p}\\
\alpha & \beta_3\;\gamma_5\;\underline{q}\\
\hline
\end{array}
\qquad
\begin{array}{|l|l|}
\hline
\alpha\,\beta_1 & \underline{b}\;\underline{p}\\
\alpha\,\beta_1 & \gamma_1\;\underline{p}\\
\alpha\,\beta_1 & \gamma_2\;\underline{q}\\
\hline
\alpha\,\beta_2 & \underline{b}\;\underline{q}\\
\alpha\,\beta_2 & \gamma_3\;\underline{p}\\
\alpha\,\beta_2 & \gamma_4\;\underline{q}\\
\hline
\alpha\,\beta_3 & \gamma_5\;\underline{p}\\
\alpha\,\beta_3 & \gamma_5\;\underline{q}\\
\hline
\end{array}
\qquad
\begin{array}{|l|l|}
\hline
\alpha\,\beta_1\;\underline{b} & \underline{p}\\
\hline
\alpha\,\beta_1\;\gamma_1 & \underline{p}\\
\hline
\alpha\,\beta_1\;\gamma_2 & \underline{q}\\
\hline
\alpha\,\beta_2\;\underline{b} & \underline{q}\\
\hline
\alpha\,\beta_2\;\gamma_3 & \underline{p}\\
\hline
\alpha\,\beta_2\;\gamma_4 & \underline{q}\\
\hline
\alpha\,\beta_3\;\gamma_5 & \underline{p}\\
\alpha\,\beta_3\;\gamma_5 & \underline{q}\\
\hline
\end{array}
$$

Figure 1.8: The matrix term representation of (1.8) and its prefix partitions.

variables. The problem is, how do we go about proving there is a bound for the number of literals and/or variables in any such clause. We solve this problem by introducing the notion of *prefix partitioning* which induces a nested division of any clause over which an inductive argument establishes the required cardinality bounds. We develop our argument for sets of terms and not for clauses. We encode clauses as sets of terms all with the same length, namely $m + 1$. The clause (1.8) is encoded by the first set in Figure 1.8, depicted as a matrix. The symbols $\underline{b}$, $\underline{p}$ and $\underline{q}$ are new. $\underline{p}$ and $\underline{q}$ are so-called predicate constants that are uniquely associated with the predicate symbols P and Q. $\underline{b}$ is the blank symbol for filling rows so that the predicate constants occur all in the last column. Our goal is to exhibit the existence of a bound for the height of any such matrix.

Prefix partitioning divides any prefix stable clause according to common prefixes of the same length. Figure 1.8 depicts the prefix partitions by common prefixes of length zero, one, two and three, respectively. The first two partitions by common prefixes of length zero and one coincide and are the entire matrix, because every term begins with [] and α, respectively. Figure 1.9 combines the three prefix partitions in one picture, and more important, it visualises the tree-like division of any prefix stable clause. This division also guides our induction proof. The literal and variable bounds we give are m-story exponential functions.

1.5 An ordered refinement

It is not our intention to use unrefined resolution and condensing for automating modal inference. The generality of the decidability result leaves us lots of room for experimenting with different refinements for improved performance. We will not attempt to give optimal refinements. Very many refined procedures exists, for example hyper-resolution, semantic resolution and lock resolution, to mention just a few. Bachmair and Ganzinger (1994, 1997) have developed a powerful and very general theory of ordered resolution that accommodates most of the familiar procedures, including the ones just mentioned. Their theory admits a strong notion of redundancy which encompasses most known simplification and deletion rules. Based on the Bachmair-Ganzinger method of saturation up to redundancy we will define an ordered E-resolution calculus for finitely based equational theories E. The spin-

α	β_1	$\underline{b}$	$\underline{p}$
α	β_1	γ_1	$\underline{p}$
α	β_1	γ_2	$\underline{q}$
α	β_2	$\underline{b}$	$\underline{q}$
α	β_2	γ_3	$\underline{p}$
α	β_2	γ_4	$\underline{q}$
α	β_3	γ_5	$\underline{p}$
α	β_3	γ_5	$\underline{q}$

Figure 1.9: The nested prefix partition of (1.8).

off is, condensing, normalisation and other theory specific simplification rules are instances of the general redundancy notion. The difficulty of devising a specialisation of ordered resolution for a particular theory is, finding a suitable ordering $\prec$ on atoms that will work. An important prerequisite for the completeness of ordered E-resolution is the compatibility of $\prec$ with E. It requires that for any atoms A, A', B, B',

if $A \prec B$, A and A' are E-equivalent and B and B' are E-equivalent, then $A' \prec B'$.

The theories we concentrate on are determined by combinations of right identity and associativity which form convergent rewrite systems based on the rules

$$[x\underline{e}] \Rightarrow x \quad \text{and}$$
$$[x(\alpha \circ \beta)] \Rightarrow [x\alpha\beta].$$

The rules are oriented by the lexicographic path ordering $\prec_{\mathrm{lpo}}$ defined from right to left with $\underline{e} \prec \circ \prec [\cdot,\cdot]$. The ordering $\prec_{\mathrm{lpo}}$ is not compatible with either of associativity or identity.[3] We will use the following ordering which is compatible with the relevant theories: For any atoms A and B,

$$A \prec^{\times}_{\mathrm{lpo}} B \quad \text{iff} \quad \text{(i) } N_E(A) \prec_{\mathrm{lpo}} N_E(B) \text{ (the ordering of the normal forms) or}$$
$$\text{(ii) } N_E(A) = N_E(B) \text{ and } A \prec_{\mathrm{lpo}} B.$$

Apart from achieving greater efficiency, we hoped that an elementary ordered resolution decision procedure can be devised for associative path logics (corresponding to transitive modal logics). After all, most of the known solvable fragments of first-order logic (including the relational and semi-functional translations of many decidable modal logics) can be decided by some form of ordered resolution (Joyner Jr. 1976, Fermüller et al. 1993, Hustadt 1997). In many cases an essential ingredient is an encoding and often the encoding uses the method of renaming (or forming definitional forms). Transformation to clausal form by renaming has polynomial as opposed to exponential overhead, which the naive transformation via negation normal form has.

[3]In Section 7.2 and the subsequent sections the index in $\prec_{\mathrm{lpo}}$ will be suppressed.

Renaming will be considered on the modal level. A given modal formula φ is reformulated by introducing names for subformulae of φ. For example, renaming produces for

$$\rho_3 = \underset{a_1}{\Box p} \to \underset{a_3 a_2 a_1}{\Box\Box\Box p}$$

the formula

$$\begin{aligned}\rho_3^d = [&(a_1 \leftrightarrow \Box p) \wedge \Box(a_1 \leftrightarrow \Box p) \wedge \Box^2(a_1 \leftrightarrow \Box p)\\ &\wedge (a_2 \leftrightarrow \Box a_1) \wedge \Box(a_2 \leftrightarrow \Box a_1)\\ &\wedge (a_3 \leftrightarrow \Box a_2)]\\ &\to (a_1 \to a_3).\end{aligned}$$

It is called a modal definitional form of ρ_3. a_1, a_2, a_3 are the new variables introduced as indicated: a_1 for both occurrences of $\Box p$, a_2 for $\Box a_1$ and a_3 for $\Box a_2$. The conjunction $(a_1 \leftrightarrow \Box p) \wedge \Box(a_1 \leftrightarrow \Box p) \wedge \Box^2(a_1 \leftrightarrow \Box p)$ is abbreviated by $\Box^{(2)}(a_1 \leftrightarrow \Box p)$. It constitutes the definition of $\Box p$ and is determined by the maximal modal depth of any occurrence of $\Box p$ in ρ_3. Definitions of the general form $\Box^{(n)}\psi$ can often be simplified. In KT, because $\Box p \to p$ is an axiom schema, $\Box^n\psi$ implies $\Box^{(n)}\psi$, meaning the definition simplifies to $\Box^n\psi$. For the schema $4 = \Box p \to \Box\Box p$ the definition simplifies to $\Box^{(1)}\psi$ and in *S4* which includes both schemas, the definition can be simplified to just $\Box\psi$.

Definitional forms are not unique. We consider forms of renaming producing so-called *prefix ordered* clauses. Let Es denote a disjunction of unary literals all with the argument s. A prefix ordered clause has the form

$$E_0 s \vee E_1[su_1] \vee E_2[su_1u_2] \vee \ldots \vee E_n[su_1 \ldots u_n],$$

where each u_i is either a variable or constant. A prefix ordered clause is a clause of basic path logic with the set of its terms being totally ordered by the proper subterm ordering. Under the empty theory and associativity this form is preserved by ordered E-resolution with normalisation. All the variables of any prefix ordered clause occur in the term $[su_1 \ldots u_n]$. Given that a term depth bound (A) exists, the existence of a bound (B) for the maximum number of variables in any condensed clause is immediate, and decidability follows, in particular, for the basic path logic.

Introducing new names for *each* non-literal subformula of the given φ generates clauses of the form

$$E_0[] \vee E_1[\alpha_1] \vee E_2[\alpha_1\alpha_2] \vee \ldots \vee E_n[\alpha_1\alpha_2 \ldots \alpha_n] \vee E_{n+1}[\alpha_1\alpha_2 \ldots \alpha_n u]$$

for u a variable or constant and $0 \leq n \leq m$ for some m. This form is also preserved by ordered E-resolution under the empty theory and associativity, and decidability for basic path logic is evident. Moreover, an improved bound for the number of literals of any condensed clause of this form exists and is a polynomial in the given m and k, as opposed to the m-story exponential bound for unrefined resolution.

Devising a resolution decision procedure for associative path logics amounts to finding simplification rules and other strategies that prevent terms to grow indefinitely. We will discuss this problem without presenting a satisfactory solution.

Chapter 2

Modal logic and the functional translation

This chapter provides some background on propositional modal logic and its standard relational semantics and it introduces the functional translation.

Historically, the functional translation approach appeared simultaneously and independently in the late eighties in a number of dissertations and publications, namely Ohlbach (1988a, 1988b, 1991), Fariñas del Cerro and Herzig (1988, 1989), Herzig (1989), Auffray and Enjalbert (1992) and Zamov (1989). Except for Zamov who considers the logic $S4$ the functional translation method is defined for quantified modal logic. Survey papers are by Ohlbach (1993b) and Fariñas del Cerro and Herzig (1995).

The functional translation approach is a semantic approach (as opposed to a syntactic approach which encodes Hilbert-style axiomatisations, see Morgan 1976). It is based on an alternative functional semantics of modal logic, which evolves naturally from the standard relational semantics. Starting with a brief review in Section 2.1 of the essential background on modal logics, the relational semantics and the relational translation, Sections 2.2 and 2.3 introduce the functional semantics and its correspondence theory. Section 2.4 defines the functional translation mapping and presents a syntactic proof of its completeness. The functional translation mapping embeds modal logics in a lattice of logics we call non-optimised path logics. The weakest logic in this lattice is called basic non-optimised path logic and will be defined formally in Section 2.5. One of its pleasant properties important in all subsequent chapters is prefix stability. Finally, Section 2.6 is concerned with the observation that under the functional perspective the basic modal logic is the serial modal logic KD rather than the logic K.

2.1 Modal logics and the relational semantics

The reader is assumed to be familiar with propositional modal logic. Good introductory textbooks on modal logic are Chellas (1980), Goldblatt (1987) and Hughes and Cresswell (1996, 1984). There are numerous survey papers, including Bull and Segerberg (1984) and Fitting (1993). The decisive survey on correspondence theory is van Benthem (1984). The purpose of this section is to review briefly the basic notions of normal modal logics, including the relational translation.

K and $K_{(m)}$ are the basic modal logics we consider. The language of basic modal logic is that of propositional logic plus one or more unary modal operators. Propositions are interpreted as unary predicates of worlds and the modality determines a restricted form of quantification on a binary relation. In applications like knowledge representation quantification is over more than one binary relation. Description logics are multi-modal logics with several modalities (Schild 1991, van der Hoek and de Rijke 1995). The modalities are $\Diamond$, for uni-modal logics, and $\Diamond_i$ with i in some index set, for multi-modal logics. Since uni-modal logic is the instance of multi-modal logic in which the index set is a singleton set, we give the definition of the basic multi-modal logic $K_{(m)}$. Let V be a finite set of propositional variables $p, q, r, \ldots$ and let m be a natural number. A *modal formula* is defined inductively by:

(i) Every propositional variable p in V is a modal formula.

(ii) $\bot$ (false) is a modal formula.

(iii) $\varphi \to \psi$ is a modal formula, when both φ and ψ are modal formulae.

(iv) $\Diamond_i \varphi$ is a modal formula, when $1 \leq i \leq m$ and φ is a modal formula.

The Boolean connectives $\neg$, $\vee$, $\wedge$, $\leftrightarrow$ and $\top$ (truth) are defined as usual. The modal box operators $\Box_i$ are duals of the respective diamond operators. By definition, $\Box_i \varphi = \neg \Diamond_i \neg \varphi$, for any modal formula φ. A *modal schema* is a modal formula representing a collection of modal formulae as determined by the schema. For example, by the schema $\Diamond_i p \to p$ we mean the collection of formulae $\{\Diamond_i \varphi \to \varphi \mid \varphi \text{ is any modal formula}\}$. A *modal axiom* is an instance of a modal schema.

A *normal modal logic* is defined by a set of modal formulae which includes all propositional tautologies over some modal language and the schema

$$K \qquad \Box_i (p \to q) \to (\Box_i p \to \Box_i q)$$

for each modality, and which is closed under modus ponens and the rule of *necessitation*:

$$MP \qquad \text{if } \vdash p \text{ and } \vdash q \text{ then } \vdash p \to q$$

$$N \qquad \text{if } \vdash p \text{ then } \vdash \Box_i p.$$

K and $K_{(m)}$ are the weakest normal modal logics. In general, normal modal logics are extensions of the logics K and $K_{(m)}$ with additional schemas like:

$$D \qquad \Box_i p \to \Diamond_i p$$

$$T \qquad \Box_i p \to p$$

$$B \qquad p \to \Box_i \Diamond_i p$$

$$4 \qquad \Box_i p \to \Box_i \Box_i p$$

and additional rules of the form

$$\text{if } \vdash \varphi_1 \text{ and } \ldots \text{ and } \vdash \varphi_n \text{ then} \vdash \varphi. \tag{2.1}$$

Throughout the thesis we let Σ be a finite set of such schemas and rules. By $K\Sigma$ and $K_{(m)}\Sigma$ we denote the smallest normal modal logics containing the schemas in Σ, closed under the additional rules in Σ. We also use the notational convention by which *KT4*, for example, denotes the smallest extension of K in which both T and 4 are schemas. Alternative names for *KT4* and *KTB4* are *S4* and *S5*. A modal formula φ is a *theorem* of K, $K_{(m)}$ or their extensions iff φ can be derived from the axioms by using the rules of the appropriate logic.

The standard semantics of normal propositional modal logics, known as the *Kripke semantics* or *possible world semantics*, is given in terms of relational structures called *frames*. A frame of a multi-modal logic is a pair $\mathcal{F} = (W, \{R_i\}_i)$ of a non-empty set of worlds W and a family of binary relations R_i on W. The R_i are the accessibility relations that determine the truth of modal formulae in possible worlds. The defining class of frames of a modal logic determines, and is determined by, a corresponding class of models. A (*relational*) *model* is a pair $\mathcal{M} = (\mathcal{F}, \iota)$ of a frame $\mathcal{F}$ and a *valuation* mapping ι. ι is a function that assigns subsets of W to atomic propositional variables. The model is said to be *based on the frame* $(W, \{R_i\}_i)$. *Truth* (or *validity*) in any model $\mathcal{M} = (W, \{R_i\}_i, \iota)$ and any world $x \in W$ is defined inductively by:

$$
\begin{array}{ll}
\mathcal{M}, x \models p & \text{iff } x \in \iota(p) \\
\mathcal{M}, x \not\models \bot & \\
\mathcal{M}, x \models \varphi \rightarrow \psi & \text{iff } \mathcal{M}, x \models \varphi \text{ then } \mathcal{M}, x \models \psi \\
\mathcal{M}, x \models \Diamond_i \varphi & \text{iff } (x, y) \in R_i \text{ and } \mathcal{M}, x \models \varphi \text{ for some } y \in W.
\end{array}
$$

A modal formula is *valid in a frame* iff it is valid in all models based on the frame. The basic modal logic $K_{(m)}$ is completely determined by the class of all frames $(W, \{R_i\}_i)$.

Normal modal logics can be studied systematically by considering the classes of frames they define. In general, these are subclasses of the class of all frames which define the basic modal logics. A normal modal logic $K_{(m)}\Sigma$ is said to be *sound* (respectively *complete*) *with respect to a class of frames* iff for any modal formula φ, any frame in the class validates φ if (respectively iff) φ is a theorem in $K_{(m)}\Sigma$. A normal modal logic is said to be *complete* iff it is complete with respect to some class of frames. In this thesis the logic *KD* has special importance. *KD* is complete with respect to the class of frames in which R is total (or serial). Figure 2.1 lists the first-order correspondence properties that restrict the classes of frames for extensions $K\Sigma$ for a selection of common schemas.

Not every modal schema has an equivalent *first-order* frame property. A class of frames comprising of all frames satisfying a set of first-order conditions is said to be an *elementary* class. A modal logic is *first-order definable* if it is characterised by an elementary class of frames. In Chapter 3 we will consider the extension of K with McKinsey's schema

$$M \qquad \Box_i \Diamond_i p \rightarrow \Diamond_i \Box_i p.$$

The logic *KM* is not determined by any elementary class of frames. Such logics are said to be *essentially second-order* modal logics. *KM* is also not canonical, which is to say it is not validated by its canonical frame (constructed with maximally consistent sets). Nevertheless, Fine (1975) shows *KM* is complete, and it has the finite model property and is decidable. M appears to be the smallest example of a schema which is not Sahlqvist. Some normal modal logics cannot be associated with any class of characteristic *frames* and these are said to be *incomplete*. In this thesis we will not consider any incomplete logics.

The *relational translation* of modal logics imitates their relational semantics. In general, the relational translation is a mapping Π_r of modal formulae into second-order logic. For any propositional modal logic $K_{(m)}\Sigma$, Π_r is a function given by:

$$
\Pi_r(\varphi) = \begin{cases} \forall P_1 \ldots P_n \; \forall x \; \pi_r(\varphi, x) & \text{if } \varphi \text{ is an axiom schema in } \Sigma \text{, and} \\ \forall x \; \pi_r(\varphi, x) & \text{if } \varphi \text{ is not an axiom schema.} \end{cases}
$$

D	$\Box p \to \Diamond p$
	totality/seriality: $\forall x\ \exists y\ R(x,y)$
T	$\Box p \to p$
	reflexivity: $\forall x\ R(x,x)$
B	$p \to \Box\Diamond p$
	symmetry: $\forall xy\ (R(x,y) \to R(y,x))$
4	$\Box p \to \Box\Box p$
	transitivity: $\forall xyz\ ((R(x,y) \wedge R(y,z)) \to R(x,z))$
4^2	$\Box^2 p \to \Box^3 p$
	$\forall xyzu\ ((R(x,y) \wedge R(y,z) \wedge R(z,u)) \to \exists v\ (R(x,v) \wedge R(v,u)))$
5	$\Diamond p \to \Box\Diamond p$
	euclideanness: $\forall xyz\ ((R(x,y) \wedge R(x,z)) \to R(y,z))$
G	$\Diamond\Box p \to \Box\Diamond p$
	confluence: $\forall xyz\ ((R(x,y) \wedge R(x,z)) \to (\exists w\ R(y,w) \wedge R(z,w)))$
$D1$	$\Box(\Box p \to q) \vee \Box(\Box q \to p)$
	$\forall xyz\ ((R(x,y) \wedge R(x,z)) \to (R(z,y) \vee R(y,z)))$
Funct	$\Diamond p \to \Box p$
	single-valuedness: $\forall xyz\ ((R(x,y) \wedge R(x,z)) \to y = z)$
w. dens.	$\Box\Box p \to \Box p$
	weak density: $\forall xy\ (R(x,y) \to (\exists z\ R(x,z) \wedge R(z,y)))$
Mk	$(\Box p \wedge q) \to \Diamond(\Box\Box p \wedge \Diamond q)$
	$\forall x \exists y\ (R(x,y) \wedge R(y,x) \wedge (\forall z\ R^2(y,z) \to R(x,z)))$

Figure 2.1: Relational correspondence properties

The P_i are unary predicate symbols uniquely associated with the propositional variables p_i occurring in φ as defined by the auxiliary function π_r (often denoted by ST). π_r is a function from modal formulae and worlds to first-order formulae given by:

$$\begin{aligned}\pi_r(p_i, x) &= P_i(x)\\ \pi_r(\perp, x) &= \perp\\ \pi_r(\varphi \to \psi, x) &= \pi_r(\varphi, x) \to \pi_r(\psi, x)\\ \pi_r(\Diamond_i \varphi, x) &= \exists y\ R_i(x, y) \wedge \pi_r(\varphi, y).\end{aligned}$$

For sets of schemas the relational translation is given by

$$\Pi_r(\Sigma) = \bigwedge_{\varphi \in \Sigma} \Pi_r(\varphi),$$

and for rules, which have the general form (2.1) Π_r is defined by

$$(\Pi_r(\varphi_1) \wedge \ldots \wedge \Pi_r(\varphi_n)) \to \Pi_r(\varphi),$$

The translation of schemas and rules (with Π_r) yields second-order formulae with universally quantified predicate variables P_i. For some schemas and rules the translations are equivalent to a first-order formula, for example, the translations for the schema K and necessitation are first-order tautologies. Since $K_{(m)}$ is a complete modal logic it follows, φ is a theorem in $K_{(m)}$ iff the first-order equivalent of its translation $\Pi_r(\varphi)$ is a first-order theorem. We say the relational translation is *sound and complete* for $K_{(m)}$. For devising inference methods based on the possible worlds semantics it is important to know the characteristic frame properties for the axioms in Σ. Because we want to implement modal deduction in first-order theorem provers, we are merely interested in schemas which can be reduced to first-order logic. For complete and first-order definable modal logics, we will regard $\Pi_r(\Sigma)$ as denoting a set of first-order correspondence properties associated with Σ and

$$\varphi \text{ is a theorem in } K_{(m)}\Sigma \quad \text{iff} \quad \Pi_r(\Sigma) \to \Pi_r(\varphi) \text{ is a first-order theorem.}$$

Not every complete modal logic can be defined by first-order properties of the accessibility relations. So, in general, for any complete modal logics $K_{(m)}\Sigma$,

$$\varphi \text{ is a theorem in } K_{(m)}\Sigma \quad \text{iff} \quad \Pi_r(\Sigma) \to \Pi_r(\varphi) \text{ is a second-order theorem.}$$

2.2 The functional semantics

The *functional semantics* can be viewed as arising from the relational semantics by a reformulation of relations in terms of sets of functions. The fundamental idea is that any binary relation can be defined by a set of (partial or total) functions.

Theorem 2.2.1 For any binary relation R on a non-empty set W there is a set AF_R of partial functions α from W to W such that the following holds:

$$\forall xy\ (R(x, y) \ \leftrightarrow\ (\exists \alpha\ \alpha \in AF_R \wedge \alpha(x) = y)). \tag{2.2}$$

Proof. Binary relations can be regarded as many-valued functions. Any relation R is a function mapping any element x in W to the set $R''(x)$ of R-images of x. By the axiom of choice, for any family F of non-empty sets there exists a function $\sigma : F \longrightarrow \bigcup F$ assigning to any non-empty set Y in F an element of Y, that is, $\sigma(Y) \in Y$. σ is the choice (or selector) function. It is not defined on empty sets. If SF is the set of all choice functions for F then $Y = \{\sigma(Y) \mid \sigma \in SF\}$. In particular, let SF denote the set of all choice functions $\sigma : F \longrightarrow W$ for the family F ($\subseteq \mathbf{2}^W$) of sets $R''(x)$, for x in W, with $R''(x) \neq \emptyset$. Then, $R''(x) = \{\sigma(R''(x)) \mid \sigma \in SF\}$. Observe that the composition $R'' \circ \sigma^1$ of R'', regarded as a many-valued function, and any choice function σ is a partial function on the set of possible worlds. Now, let AF_R be the set of such partial functions, that is, let $AF_R = \{R'' \circ \sigma \mid \sigma \in SF\}$. Then $R''(x) = \{\alpha(x) \mid \alpha \in AF_R\}$, which defines each accessibility relation R over a set W of worlds as a union of (partial) functions in AF_R. □

We call any set AF_R such that (2.2) holds a set of *accessibility functions defining* R, or a *defining function set* for R.

In order to avoid quantification over function symbols (in the first-order translation), like in (2.2), we prefer to use a list notation in which any term

$$\alpha(x) \quad \text{is rewritten as} \quad [x\alpha].$$

$[\cdot,\cdot]$ denotes the *functional application* operation defined to be a binary (non-associative) function from $W \times \mathrm{Par}(W,W)$ to W, where $\mathrm{Par}(W,W)$ is the set of all partial functions over W. Thus, complex terms of the form $\alpha_m(\ldots\alpha_2(\alpha_1(x)))$ become terms of the form

$$[[[[x\alpha_1]\alpha_2]\ldots]\alpha_m].$$

We adopt the convention of omitting all parentheses except for the outer pair and write

$$[x\alpha_1\alpha_2\ldots\alpha_m]$$

instead. Reformulating (2.2), any binary relation R is then defined by a set AF_R of functions and the operation $[\cdot,\cdot]$ if

$$\forall xy\ (R(x,y) \ \leftrightarrow\ \exists\alpha\ (\alpha \in AF_R \wedge [x\alpha] = y)). \tag{2.3}$$

α is now a first-order variable in a many-sorted language, as will be made precise in Section 2.4 below.

Returning to the theorem, we note that the proof does not uniquely determine AF_R. It exhibits for (2.2) to hold that, AF_R is a suitable (defining) subset of the set

$$\{\alpha \in \mathrm{Par}(W,W) \mid \exists x \in W\ R(x,[x\alpha])\}.$$

This definition is very general and also allows us to choose AF_R to be a suitable subset of the set of all total functions α from the *domain of* R to W given by

$$\{\alpha \in \mathrm{dom}(R) \longrightarrow W \mid \forall x \in W\ R(x,[x\alpha])\}.$$

To emphasise the point, as an illustration consider the relation R of Figure 2.2. There are eight different ways to accommodate this relation in terms of sets of two functions $\{\alpha,\beta\}$,

[1] Contrary to common convention we use the notation $\circ$ like relational composition: $(f\circ g)(x) = g(f(x))$.

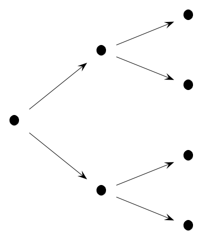


Figure 2.2: A sample relation R

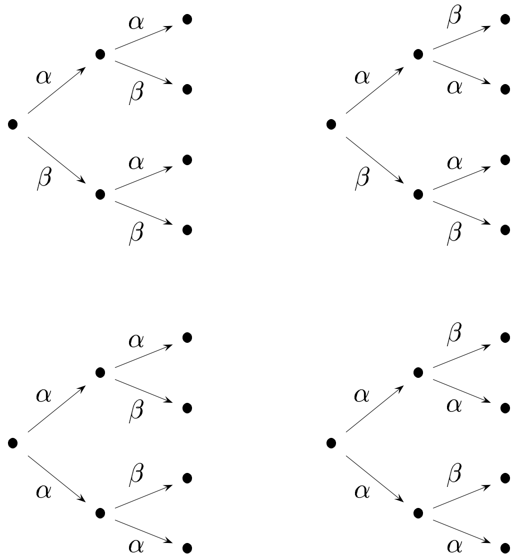


Figure 2.3: Four different functional encodings of R

four of which are depicted in Figure 2.3. α and β are total functions from $\mathrm{dom}(R)$ to W. Since the relation is not total, all these α's and β's are partial functions from W to W. Using partial functions from W to W there are even more possible decompositions for a relation. Figure 2.4 shows an extreme case where the functions $\alpha_1, \ldots, \alpha_6$ are as partial as possible and map only one world to another world.

As illustrated in the Preview, for total relations it is always possible to define the relation in terms of a set of total functions from W to W. For example, the relation R of Figure 2.5 is total and all functions α_i are total.

For the general case that R is not total, no set AF_R exists that is a set of *total* functions from W to W. The problem with this is that the target logic for the functional translation

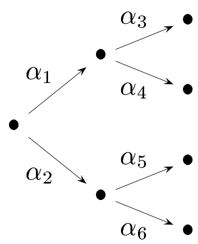


Figure 2.4: R encoded by a set of 'most partial' functions

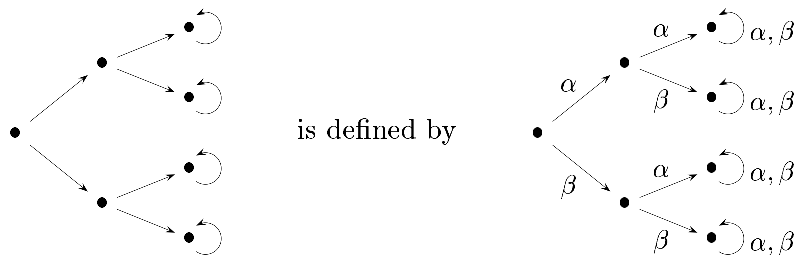


Figure 2.5: R encoded by the total functions α and β

has no built-in facilities for dealing with partial functions. $[x\alpha]$ is a first-order term and will always have an interpretation. Following the standard solution we adjoin a special element $\perp$ to W thus obtaining $W^{\perp}$ and encode every partial function α as a total function which maps the elements for which α is not defined to the new 'undefined' world $\perp$. Accordingly, any formula in which such an 'undefined' situation occurs is translated into a conditional formula. Instead of formulating the condition 'the function α is defined for x' by an inequality $[x\alpha] \neq \perp$, we introduce a designated predicate de_R, called the *dead-end* predicate, which is defined by

$$\forall x\ (de_R(x)\ \leftrightarrow\ \forall \alpha\ (\alpha \in AF_R \rightarrow [x\alpha] = \perp)). \tag{2.4}$$

Theorem 2.2.2 Let R be any binary relation on W. Let $W^{\perp} = W \cup \{\perp\}$. The following defines R by a set AF_R of *total* functions α from $W^{\perp}$ to $W^{\perp}$: For any $x, y \in W$

$$\forall xy\ (R(x,y)\ \leftrightarrow\ (\neg de_R(x) \wedge \exists \alpha\ \alpha \in AF_R \wedge [x\alpha] = y)) \tag{Def-R}$$

and de_R defined by (2.4).

Proof. We adapt the proof of the previous theorem. Let de_R be the set $\{x \mid \neg\exists y\ R(x,y)\}$. Let AF_R be the set of functions α defined by

$$\alpha(x) = \begin{cases} (R'' \circ \sigma)(x) & \text{if } x \notin de_R \text{ and } \sigma \text{ is any choice function} \\ \perp & \text{otherwise.} \end{cases}$$

The α are well-defined (total) functions from $W^{\perp}$ to $W^{\perp}$.

Let $x, y \in W$ be arbitrary. $(x,y) \in R$ iff $y \in R''(x)$ iff $x \notin de_R$ and $y \in R''(x)$ iff $x \notin de_R$ and for any choice function σ we have, $(R'' \circ \sigma)(x) = y$ iff $x \notin de_R$ and there is a function α in AF_R such that $\alpha(x) = y$. □

(Def-R) defines any binary relation by a set AF_R of total functions and a special set de_R of worlds, those without R-successor worlds. The right hand side of (Def-R) reads, if x is not a dead-end in R then there is a total function α, which maps x to y. If R is total then de_R coincides with W and (Def-R) simplifies to (2.3).[2]

Now we are ready to define the functional semantics formally. A *functional frame* is a quadruple

$$\mathcal{F} = (W, \{de_i\}_i, \{AF_i\}_i, [\cdot,\cdot]),$$

[2] There is a mistake in (5.2) of Ohlbach and Schmidt (1997), the '$\rightarrow$' should be an '$\wedge$'.

with W a non-empty set of worlds, $\{de_i\}_i$ a family of subsets of W, $\{AF_i\}_i$ a family of sets of *total* functions from W to W and $[\cdot,\cdot] : W \times \bigcup_i AF_i \longrightarrow W$ the functional application operation. Each AF_i is called a set of *accessibility functions.* A *functional model* is a pair $(\mathcal{F}, \iota)$ of a functional frame and a valuation mapping of propositional variables to subsets of W. *Truth* (or *validity*) in any functional model $\mathcal{M}$ and any world $x \in W$ is inductively defined by:

$$
\begin{array}{ll}
\mathcal{M}, x \models p & \text{iff } x \in \iota(p) \\
\mathcal{M}, x \not\models \bot & \\
\mathcal{M}, x \models \varphi \to \psi & \text{iff } \mathcal{M}, x \models \varphi \text{ then } \mathcal{M}, x \models \psi \\
\mathcal{M}, x \models \Diamond_i \varphi & \text{iff, } x \notin de_i \text{ and there is an } \alpha \in AF_i \text{ such that } \mathcal{M}, [x\alpha] \models \varphi.
\end{array}
$$

We will prove a general completeness result which derives from completeness with respect to relational frames by the relationship (Def-R). A functional model can be defined from a relational model $\mathcal{M} = (W, \{R_i\}_i, \iota)$ by letting $W^\perp$ and each de_i and AF_i be as in Theorem 2.2.2. The structure

$$\mathcal{M}^f = (W^\perp, \{de_i\}_i, \{AF_i\}_i, [\cdot,\cdot], \iota)$$

is a well-defined functional model. We say $\mathcal{M}^f$ is a *functional model based on the relational model* $\mathcal{M}$.

Theorem 2.2.3 Let φ be a modal formula and let $\mathcal{M} = (W, \{R_i\}_i, \iota)$ be a relational model. For any $x \in W$,

$$\mathcal{M}, x \models \varphi \quad \text{iff} \quad \mathcal{M}^f, x \models \varphi.$$

Proof. The proof is by induction on the structure of the formula φ. Since the valuation assignments of the two models are identical, the base case of the induction is trivial. We also omit the proofs for φ being constructed with the propositional connectives and give a proof where φ has the form $\Diamond_i \psi$. Using Theorem 2.2.2 we have that $\mathcal{M}, x \models \Diamond_i \psi$ iff there is a $y \in W$ such that $(x, y) \in R_i$ and $\mathcal{M}, y \models \psi$, iff there is a $y \in W$ such that $x \notin de_i$ and $\exists \alpha \in AF_i\ [x\alpha] = y$ and $\mathcal{M}, y \models \psi$, iff there is a $y \in W$ such that $x \notin de_i$ and $\exists \alpha \in AF_i$ and $\mathcal{M}, [x\alpha] \models \psi$, iff $x \notin de_i$ and $\exists \alpha \in AF_i$ such that $\mathcal{M}, [x\alpha] \models \psi$. □

A particularly important construction is the construction of the class of so-called *maximal functional models* (or frames). Any relational model $\mathcal{M}$ determines a unique maximal functional model, denoted by $\mathcal{M}^m$, which is defined as follows. $\mathcal{M}^m$ is the functional model in which the function sets are defined by

$$AF_i^m = \{\alpha \in W^\perp \longrightarrow W^\perp \mid \forall x \in W\ R_i(x, [x\alpha]) \text{ or } [x\alpha] = \bot\}, \tag{2.5}$$

the set of *all* total functions which define R. A maximal functional frame is in fact the *largest* functional frame defining a relational frame.

Corollary 2.2.4 Let $\mathcal{M}$ be any relational model. For any modal formula φ and any $x \in W$,

$$\mathcal{M}, x \models \varphi \quad \text{iff} \quad \mathcal{M}^m, x \models \varphi.$$

Proof. By Theorem 2.2.3. □

Van Benthem (1993) uses *patching* to construct maximal functional models. Given a functional model define AF_i^p to be an extension of AF_i that is closed under the principle of patching: Any maximally satisfiable union of domain restrictions of functions inside AF_i^p must itself be a function inside AF_i^p. Formally, for any function α from W to W and any $x \in W$,

$$\text{if } \exists \beta \in AF_i^p \text{ and } [x\alpha] = [x\beta] \text{ then } \alpha \in AF_i^p.$$

Clearly, AF_i^p and AF_i^m coincide.

Going in the other direction we can construct a *relational model* $\mathcal{M}^r$ *from a functional model* $\mathcal{M}$. Given that $\mathcal{M}$ is a functional model, its corresponding relational model is

$$\mathcal{M}^r = (W^r, \{R_i\}_i, \iota),$$

where $W^r = W \setminus \{\bot\}$ and for every i, $R_i = \{(x, [x\alpha]) \mid [x\alpha] \neq \bot \text{ and } \alpha \in AF_i\}$.

Theorem 2.2.5 Let φ be any modal formula and let $\mathcal{M}$ be a functional model. For any $x \in W$,

$$\mathcal{M}, x \models \varphi \quad \text{iff} \quad \mathcal{M}^r, x \models \varphi.$$

Proof. The argument is almost identical to the argument of the proof of the previous theorem. □

We can show:

Theorem 2.2.6 (i) $(\mathcal{M}^f)^r = \mathcal{M}$, when $\mathcal{M}$ is a relational model, and
(ii) $(\mathcal{M}^r)^m = \mathcal{M}$, when $\mathcal{M}$ is a maximal functional model.

By Theorems 2.2.3 and 2.2.5 completeness with respect to functional frames follows:

Theorem 2.2.7 Any modal logic $K_{(m)}\Sigma$ is complete with respect to a class of relational frames (models) iff it is complete with respect to a class of functional frames (models).

The theorem ensures that the functional translation which will be defined in the next section is sound and complete. The results 2.2.4, 2.2.5 and 2.2.6 yield a stronger completeness result which has special relevance for globalisation and the optimised functional translation:

Corollary 2.2.8 Any modal logic $K_{(m)}\Sigma$ is complete with respect to a class of relational frames (models) iff it is complete with respect to the class of maximal functional frames (models).

2.3 Correspondence theory

Based on the last two results, by using (Def-R) any relational correspondence property can be translated into a functional correspondence property. This is done by plugging the definition (Def-R) into the known relational correspondence property, for example, transitivity. Assuming seriality (which means we may use (2.3)), we replace R uniformly by $\exists\alpha\ [x\alpha] = y$ and produce:

$$
\begin{array}{lll}
 & \forall xyz\ (R(x,z) \wedge R(z,y)) \rightarrow R(x,y) & \\
 & \text{iff} \quad \forall xyz\ ((\exists\alpha\ z = [x\alpha]) \wedge (\exists\beta\ y = [z\beta])) \rightarrow (\exists\gamma\ y = [x\gamma]) & \\
 & \text{iff} \quad \forall xy\ (\exists\alpha\beta\ y = [x\alpha\beta]) \rightarrow (\exists\gamma\ y = [x\gamma]) & \\
 & \text{iff} \quad \forall xy\ \forall\alpha\beta\exists\gamma\ (y = [x\alpha\beta] \rightarrow y = [x\gamma]) & \\
(2.6) & \text{iff} \quad \forall x\ \forall\alpha\beta\exists\gamma\ [x\alpha\beta] = [x\gamma] &
\end{array}
$$

Alternatively, functional correspondence properties can be derived automatically with the SCAN algorithm of Gabbay and Ohlbach (1992) (just as the relational correspondence properties can).[3] Figure 2.6 lists the general correspondence properties for some schemas computed by SCAN (to avoid cluttering the indexes are omitted). Since SCAN is sound and by completeness with respect to functional frames (Theorem 2.2.7):

Theorem 2.3.1 The pairs (φ, ψ) of modal axioms and first-order formulae in Figure 2.6 are such that φ and ψ are logically equivalent.

For serial modalities the correspondence properties are purely equational, see Figures 2.7 and 2.8 for the Skolemised forms. The pictures illustrate the interpretations of the respective Skolem terms, and motivate the notation for the respective Skolem functions. For the schema D, the equational theory is empty. In functional frames that validate T, reflexivity is captured by the equation $[xe(x)] = x$, which we refer to as the *local (right) identity* law. Symmetry is captured by the *local (right) inverse* law $[x\alpha\, \mathit{inv}(x,\alpha)] = x$, because for any function α and any world x the local inverse function $\mathit{inv}(x,\alpha)$ maps $[x\alpha](=\alpha(x))$ to the original world x. The correspondence property (2.6) of the schema 4 is *local composability* (or *local associativity*).

For some schemas the dependency on the world variables may be removed. This operation is called *globalisation*. *Global composability* is the property

$$\forall\alpha\beta\exists\gamma\ \forall x\ [x\alpha\beta] = [x\gamma], \tag{2.7}$$

in which γ is independent of x as is made explicit in the Skolemised form

$$[x\alpha\beta] = [x(\alpha \circ \beta)].$$

$\circ$ denotes the binary Skolem function introduced for $\exists\gamma$ and defines the composition of any two functions.

Globalisation moves the world quantifier '$\forall x$' inward as far as possible. Thus global formulations logically imply the local formulations and are stronger. In general, this means satisfiability is preserved but not necessarily unsatisfiability. As the global formulations are

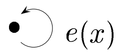
[3] In Chapter 3 and Appendix A we will consider the method of SCAN in more detail.

D	$\Box p \to \Diamond p$
	$\forall x\ \neg de(x)$
T	$\Box p \to p$
	$\forall x\ \exists \alpha\ \neg de(x) \wedge x = [x\alpha]$
B	$p \to \Box\Diamond p$
	$\forall x\ \forall \alpha\ \exists \beta\ (\neg de(x) \to \neg de[x\alpha]) \wedge (\neg de(x) \to x = [x\alpha\beta])$
4	$\Box p \to \Box\Box p$
	$\forall x\ \forall \alpha\beta\ \exists \gamma\ (\neg de(x) \wedge \neg de[x\alpha]) \to [x\alpha\beta] = [x\gamma]$
4^2	$\Box^2 p \to \Box^3 p$
	$\forall x\ \forall \alpha_1\alpha_2\alpha_3\ \exists \beta_1\beta_2\ \ (\neg de(x) \wedge \neg de[x\alpha_1] \wedge \neg de[x\alpha_1\alpha_2]) \to (\neg de[x\beta_1] \wedge [x\alpha_1\alpha_2\alpha_3] = [x\beta_1\beta_2])$
5	$\Diamond p \to \Box\Diamond p$
	$\forall x\ \forall \alpha\beta\ \exists \gamma\ (\neg de(x) \to \neg de[x\beta]) \wedge (\neg de(x) \to [x\alpha] = [x\beta\gamma])$
G	$\Diamond\Box p \to \Box\Diamond p$
	$\forall x\ \forall \alpha\beta\ \exists \gamma\gamma'\ (\neg de(x) \to (\neg de[x\alpha] \wedge \neg de[x\beta] \wedge [x\alpha\gamma] = [x\beta\gamma']))$
$D1$	$\Box(\Box p \to q) \vee \Box(\Box q \to p)$
	$\forall x\ \forall \alpha\gamma\ \exists \beta\delta\ \ (de(x) \vee \neg de[x\alpha] \vee [x\alpha] = [x\gamma\delta]) \wedge (de(x) \vee \neg de[x\gamma] \vee [x\gamma] = [x\alpha\beta]) \wedge (de(x) \vee [x\alpha] = [x\gamma\delta] \vee [x\gamma] = [x\alpha\beta]) \wedge (de(x) \vee \neg de[x\alpha] \vee \neg de[x\gamma])$
Funct	$\Diamond p \to \Box p$
	$\forall x\ \forall \alpha\beta\ de(x) \vee [x\alpha] = [x\beta]$
w. dens.	$\Box\Box p \to \Box p$
	$\forall x\ \forall \alpha\ \exists \beta\gamma\ \neg de(x) \to (de[x\beta] \wedge [x\alpha] = [x\beta\gamma])$
Mk	$(\Box p \wedge q) \to \Diamond(\Box\Box p \wedge \Diamond q)$
	$\forall x\ \exists \alpha\ \ \neg de(x) \wedge \neg de[x\alpha] \wedge (\exists \beta\ [x\alpha\beta] = x) \wedge (\forall \gamma\ de[x\alpha\gamma] \vee (\forall \delta \exists \epsilon\ [x\alpha\gamma\delta] = [x\epsilon]))$

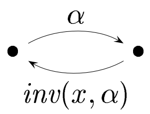
Figure 2.6: Functional correspondence properties.

$D \quad \emptyset$

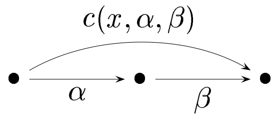
$T \quad [x\, e(x)] = x$



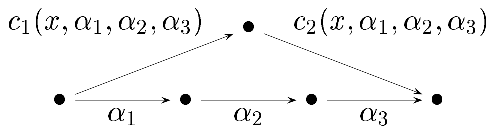
$B \quad [x\alpha\, inv(x,\alpha)] = x$



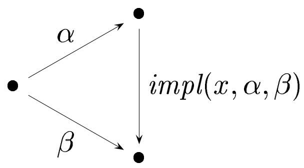
$4 \quad [x\, c(x,\alpha,\beta)] = [x\alpha\beta]$



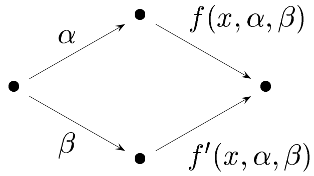
$4^2 \quad [x\alpha_1\alpha_2\alpha_3] = [x\, c_1(x,\alpha_1,\alpha_2,\alpha_3)\, c_2(x,\alpha_1,\alpha_2,\alpha_3)]$



$5 \quad [x\alpha\, impl(x,\alpha,\beta)] = [x\beta]$



$G \quad [x\alpha\, f(x,\alpha,\beta)] = [x\beta\, f'(x,\alpha,\beta)]$



$D1 \quad [x\beta\, i_1(x,\beta,\alpha)] = [x\alpha] \vee [x\alpha\, i_2(x,\alpha,\beta)] = [x\beta]$

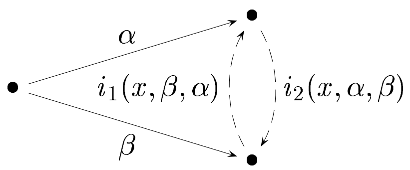
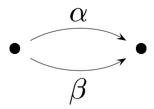
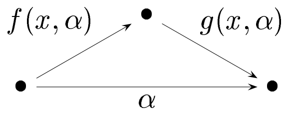


Figure 2.7: Theory equations.

Funct $[x\alpha] = [x\beta]$



w. dens. $[x\alpha] = [xf(x,\alpha)g(x,\alpha)]$



Mk $[xa(x)b(x)] = x \wedge [xa(x)\gamma\delta] = [xf(x,\gamma,\delta)]$

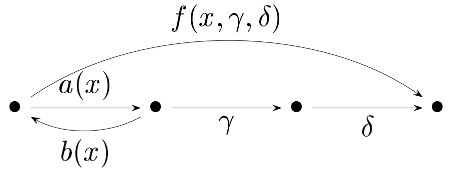


Figure 2.8: Theory equations (continued).

not definable by modal schemas we may ask whether there are schemas for which globalisation also preserves unsatisfiability, and if this is the case, are there schemas for which more can be derived modulo the global theory than modulo the local theory. In both cases the answer is yes.

Lemma 2.3.2 Let $\mathcal{M}^m$ be any maximal model. If $\mathcal{M}^m$ satisfies the property

(i) $\forall x\ \exists\alpha\ x = [x\alpha]$, then it satisfies $\exists\alpha\ \forall x\ x = [x\alpha]$, or if it satisfies

(ii) $\forall x\ \forall\alpha\beta\ \exists\gamma\ [x\alpha\beta] = [x\gamma]$, then it satisfies $\forall\alpha\beta\ \exists\gamma\ \forall x\ [x\alpha\beta] = [x\gamma]$.

Proof. (i) AF^m contains the identity function $\underline{e}$.

(ii) The composition $\alpha \circ \beta$ of any two functions α and β is a function. Hence, $\alpha \circ \beta$ is in AF^m (as it is maximal). □

By Theorem 2.3.1:

Theorem 2.3.3 Let $\Sigma \subseteq \{D, T, 4\}$. $K\Sigma$ is sound and complete with respect to the class of models satisfying the properties of Figure 2.9.

Proof. Because $Q_1y_1 \ldots Q_ny_n\forall x\ \psi$ logically implies $\forall xQ_1y_1 \ldots Q_ny_n\ \psi$ the local correspondence properties are valid in any model in which their global forms are valid. Hence, if φ is valid in a class of models as specified by Figure 2.9 then φ is a theorem in the corresponding logic $K\Sigma$.

Soundness follows by Lemma 2.3.2. □

This result is already exploited in the earliest papers on the functional translation method (see for example Herzig 1989, Zamov 1989, Ohlbach 1991, Auffray and Enjalbert 1992, Fariñas del Cerro and Herzig 1995, van Benthem 1993).

We refer to the global correspondence property of T as the *(right) identity* law. The global correspondence property of 4 can be viewed as a kind of *associativity* law (because under this law any functional variable may be expanded to a string of functional terms).

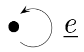
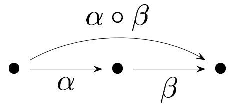
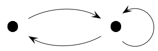
D	$de = \emptyset$	
T	$de = \emptyset \wedge x = [x\underline{e}]$	
4	$(\neg de(x) \wedge \neg de[x\alpha]) \rightarrow [x\alpha\beta] = [x(\alpha \circ \beta)]$	
$D, 4$	$de = \emptyset \wedge [x\alpha\beta] = [x(\alpha \circ \beta)]$	
$T, 4$	$de = \emptyset \wedge x = [x\underline{e}] \wedge [x\alpha\beta] = [x(\alpha \circ \beta)]$	

Figure 2.9: Global functional correspondence properties.

Globalisation is not admissible for KB. The problem is that any function with an inverse function is an injection. But for instance the model given by



cannot be defined by a set of injections. Globalisation of the identity for KDB would allow us to refute non-theorems, for example, $\rho = \Box\Diamond p \rightarrow p$, because the theory generated by $[x\alpha\, inv'(\alpha)] = x$ includes also $[x\, inv'(\alpha)\alpha] = x$.

Globalisation exemplifies that first-order equivalence is more than is required for the purposes of theorem proving. It exploits a property of maximal frames, but unfortunately there is no justification for universal globalisation. See also van Benthem (1993) and Fariñas del Cerro and Herzig (1995) on this issue.

Chapter 3 considers another instance of how completeness with respect to the class of maximal frames can be exploited. Specifically, we will see that schemas like McKinsey's schema M can be approximated by first-order conditions.

2.4 The functional translation

For technical and also for presentation reasons, we find it useful to use a sorted logic as target language for the functional translation. We employ the basic many-sorted logic with a sort hierarchy and sort declarations for function symbols (Walther 1987). In this logic, a sort symbol can be viewed as a unary predicate and denotes a subset of the domain of interpretation. A subsort declaration $S \sqsubseteq T$ is interpreted as a subset relation $\mathcal{M}(S) \subseteq \mathcal{M}(T)$, and a function sort declaration like, for example,

$$f : S_1 \times S_2 \rightarrow S \quad \text{is interpreted as} \quad \forall x, y\ S_1(x) \wedge S_2(y) \rightarrow S(f(x, y)).$$

Quantification can restrict variables to a sort.

$$\begin{array}{lll} \forall x{:}S\ \psi & \text{is interpreted as} & \forall x\ S(x) \rightarrow \psi, \text{ and} \\ \exists x{:}S\ \psi & \text{is interpreted as} & \exists x\ S(x) \wedge \psi. \end{array}$$

Any unsorted logic can be embedded in a sorted logic by introducing just one single sort denoting the entire domain. The relational translation is embedded in the sorted context by introducing one sort W for worlds and considering $\Pi_r(\Sigma) \wedge \neg\Pi_r(\varphi)$ to be a formula of sorted logic in which all variables $x, y, \ldots$ are assumed to have sort W.

For the functional translation we introduce the sort W, a family $\{AF_i\}_i$ of sorts and a special sort symbol AF^* for $\bigcup_i\{AF_i\}$. AF^* is the top functional sort and each AF_i is a subsort of AF^*. To avoid cluttering we do not always make the sort declarations explicit. Unless specified otherwise, in the remainder of this thesis the variables $x, y, z, \ldots$ are assumed to be of sort W and will be referred to as *world variables.* The *functional variables* are denoted by Greek letters $\alpha_i, \beta_i, \gamma_i, \ldots$ and are of sort AF_i. The sort of the operation $[\cdot,\cdot]$ is $W \times AF^* \longrightarrow W$. The letters $s, t, \ldots$ denote terms of sort W. They will be referred to as *world terms* or *paths.* For terms of sort AF that will often be referred to as *functional terms* we reserve the letters $u, v, w, \ldots$.

Just as the relational translation imitates the relational semantics, the functional translation imitates the functional semantics. The *functional translation* mapping is denoted by Π_f. It maps modal formulae of $K_{(m)}\Sigma$ to sorted second-order or first-order formulae as given by:

$$\Pi_f(\varphi) = \begin{cases} \forall P_1 \ldots P_n \; \forall x \; \pi_f(\varphi, x) & \text{if } \varphi \text{ is an axiom schema in } \Sigma \text{, and} \\ \forall x \; \pi_f(\varphi, x) & \text{if } \varphi \text{ is not an axiom schema.} \end{cases} \tag{2.8}$$

π_f is the auxiliary function that maps modal formulae and worlds to first-order formulae. It defines the unary predicates P_i that are uniquely associated with the propositional variables p_i in φ. For the propositional symbols and connectives, π_f is defined by

$$\begin{aligned} \pi_f(p_i, s) &= P_i s \\ \pi_f(\bot, s) &= \bot \\ \pi_f(\varphi \to \psi, s) &= \pi_f(\varphi, s) \to \pi_f(\psi, s) \end{aligned}$$

and for the modal operators, it is defined by

$$\begin{aligned} \pi_f(\Diamond_i\varphi, s) &= \begin{cases} \exists\alpha_i \; \pi_f(\varphi, [s\alpha_i]) & \text{if } \Diamond_i \text{ is a serial modality} \\ \neg de_i(s) \land \exists\alpha_i \; \pi_f(\varphi, [s\alpha_i]) & \text{otherwise} \end{cases} \\ \pi_f(\Box_i\varphi, s) &= \begin{cases} \forall\alpha_i \; \pi_f(\varphi, [s\alpha_i]) & \text{if } \Box_i \text{ is a serial modality} \\ \neg de_i(s) \to \forall\alpha_i \; \pi_f(\varphi, [s\alpha_i]) & \text{otherwise.} \end{cases} \end{aligned}$$

α_i is assumed to be a variable not occurring in s. For a set Σ of schemas and rules, Π_f is defined like Π_r:

$$\Pi_f(\Sigma) = \bigwedge_{\varphi\in\Sigma} \Pi_f(\varphi), \quad \text{and} \quad (\Pi_f(\varphi_1) \land \ldots \land \Pi_f(\varphi_n)) \to \Pi_f(\varphi).$$

Σ is assumed to be finite. Thus, the set $\Pi_f(\Sigma)$ is a finite presentation that determines a theory, in the usual sense by deductive closure. For this reason we will often speak of a theory $\Pi_f(\Sigma)$ (or later $\Upsilon\Pi_f(\Sigma)$).

We consider two examples. Take McKinsey's schema $M = \Box\Diamond p \to \Diamond\Box p$. $\Pi_f(M)$ is given by

$$\begin{aligned} &\forall P \; \forall x \; (\neg de(x) \to \forall\alpha_R(\neg de[x\alpha_R] \land \exists\beta_R \; P([x\alpha_R\beta_R])) \to \\ &\qquad (\neg de(x) \land \exists\alpha_R(\neg de[x\alpha_R] \to \forall\beta_R \; P([x\alpha_R\beta_R])))). \end{aligned}$$

When its clear from the context, as is the case in a uni-modal context, we omit the indexes of the variables. Assuming seriality, the functional translation of the formula $\rho_2 = \Box(p \rightarrow \Diamond p)$ is

$$\Pi_f(\rho_2) = \forall x \ \forall \alpha \ (P[x\alpha] \rightarrow \exists \beta \ P[x\alpha\beta]).$$

The functional translation is sound and complete as is made precise by the next theorem which is a direct consequence of Theorem 2.2.7.

Theorem 2.4.1 Provided $K_{(m)}\Sigma$ is a complete modal logic, for any modal formula φ,

(i) φ is a $K_{(m)}\Sigma$-theorem iff $\Pi_f(\Sigma) \rightarrow \Pi_f(\varphi)$ is a second-order theorem, and

(ii) φ is a $K_{(m)}\Sigma$-theorem iff $\psi \rightarrow \Pi_f(\varphi)$ is a first-order theorem, when ψ is a first-order formula such that $\psi \leftrightarrow \Pi(\Sigma)$.

By Theorem 2.3.1:

Theorem 2.4.2 The pairs (φ, ψ) of modal axioms and equational formulae in Figures 2.7 and 2.8 are such that for any modal formula φ',

$$\varphi' \text{ is a theorem in } KD\varphi \quad \text{iff} \quad \psi \rightarrow \Pi_f(\varphi') \text{ is a first-order theorem.}$$

In subsequent chapters we will study inference for serial subsystems of *S4* relying on the next theorem, which is immediate by Theorem 2.3.3.

Theorem 2.4.3 For any modal formula φ, φ is a theorem in KT, respectively $KD4$ or $S4$, iff $\Pi_f(\varphi)$ is a first-order theorem modulo the theory given by

$$\begin{array}{ll} \textit{right identity}: & [x\underline{e}] = x, \text{ respectively} \\ \textit{associativity}: & [x(\alpha \circ \beta)] = [x\alpha\beta] \end{array}$$

or both.

A step-by-step syntactic proof

Completeness of Π_f can also be proved by taking the other route in the following commutative diagram.

$$\begin{tikzcd} \begin{array}{l}\text{completeness w.r.t.}\\ \text{relational semantics}\end{array} \arrow[r] \arrow[d] & \text{completeness of } \Pi_r \arrow[d] \\ \begin{array}{l}\text{completeness w.r.t.}\\ \text{functional semantics}\end{array} \arrow[r] & \text{completeness of } \Pi_f \end{tikzcd}$$

In the presentation above we gave a semantic treatment that proceeded from the upper left corner down and across. In this section, we go across and down (as we did in Ohlbach and Schmidt 1997).[4] This alternative traversal is an example of a 'killer transformation'

[4] Here, we give a more general treatment, not restricting ourselves to the serial and uni-modal case.

as described in Ohlbach, Gabbay and Plaisted (1994). The functional translation mapping and the completeness proof is derived by a systematic syntactic method. This approach is in certain ways simpler and it is more general, in that, we can treat arbitrary Hilbert axiomatisations of non-classical logics. Arbitrary modal axioms seem not to have been studied in the functional context before.

The starting point is the relational translation of modal formulae and systems $K_{(m)}\Sigma$. In the first step, rewriting with respect to (Def-R_i) is performed. More specifically, we take the set $\Pi_r(\Sigma)$ of relational correspondence properties and add to it for each R_i the definition:

$$\text{(Def-}R_i\text{)} \qquad \forall xy\ (R_i(x,y) \leftrightarrow (\neg de_i(x) \wedge \exists \alpha_i\ [x\alpha_i] = y)).$$

We then use these equivalences for replacing all occurrences of R_i by an instance of the right-hand side of (Def-R_i). In general, adding another formula to a set of formulae may introduce inconsistencies. We have to make sure that this does not happen. This means we must prove that,

$$\Pi_r(\Sigma) \wedge \neg\Pi_r(\varphi) \text{ is satisfiable} \quad \text{iff} \quad \textstyle\bigwedge_i \text{(Def-}R_i\text{)} \wedge \Pi_r(\Sigma) \wedge \neg\Pi_r(\varphi) \text{ is satisfiable.}$$

The ($\Leftarrow$)-part of this proof is trivial, since removing a formula does not introduce inconsistencies. The ($\Rightarrow$)-part of the proof requires extending a model of the left hand side by an appropriate set of accessibility functions.

Given any relational model $\mathcal{M} = (W, \{R_i\}_i, \iota)$ we define its *functional extension* to be the model

$$\mathcal{M}^* = (W^{\perp}, \{R_i\}_i, \{de_i\}_i, \{AF_i^m\}_i, [\cdot,\cdot], \iota) \tag{2.9}$$

with $\{AF_i^m\}_i$ being the family of maximal defining function sets, that is, $\mathcal{M}^* = (\mathcal{M}^m, \{R_i\}_i)$. This extension is conservative in the sense that $\mathcal{M}^*$ does not assign different meanings to the symbols it shares with $\mathcal{M}$, namely, R_i and W. By Theorem 2.2.2, the equivalences (Def-R_i) are true in $\mathcal{M}^*$.

Theorem 2.4.4 Let $K_{(m)}\Sigma$ be a modal logic complete with respect to a class of relational models. A modal formula φ is a theorem of $K_{(m)}\Sigma$ iff

$$\Pi_r(\Sigma) \wedge \neg\Pi_r(\varphi) \wedge \textstyle\bigwedge_i \text{(Def-}R_i\text{)} \tag{2.10}$$

is unsatisfiable.[5]

Proof. For $K_{(m)}\Sigma$ assumed to be complete, the relational translation is sound and complete and we have, φ is a $K_{(m)}\Sigma$-theorem iff $\Pi_r(\Sigma) \wedge \neg\Pi_r(\varphi)$ is unsatisfiable. It suffices to show that every model $\mathcal{M}$ of $\Pi_r(\Sigma) \wedge \neg\Pi_r(\varphi)$ can be extended to a model $\mathcal{M}'$ of both $\Pi_r(\Sigma) \wedge \neg\Pi_r(\varphi)$ and each (Def-R_i). The extension that will do the job, is the functional extension $\mathcal{M}^*$ of $\mathcal{M}$. □

We can add to (2.10) other declarations and formulae provided they are true in $\mathcal{M}^*$, or, there is another extension of $\mathcal{M}^*$ which makes the additional definitions true.

[5] This theorem is the general non-serial and multi-modal version of the functional extension theorem of Ohlbach and Schmidt (1997, Theorem 4.1).

Corollary 2.4.5 Let $K_{(m)}\Sigma$ be a complete modal logic, and let Δ be a set of sort declarations and formulae free of the predicates R_i. A modal formula φ is a $K_{(m)}\Sigma$-theorem iff

$$\Pi_r(\Sigma) \wedge \neg\Pi_r(\varphi) \wedge \textstyle\bigwedge_i (\text{Def-}R_i) \wedge \Delta$$

is unsatisfiable, provided the declarations and formulae in Δ hold in the functional extension $\mathcal{M}^*$ of a relational model $\mathcal{M}$, or $\mathcal{M}^*$ has a conservative extension that satisfies Δ.

Consider the transformation of the quantificational patterns

$$\exists y\ (R_i(x,y) \wedge \pi_r(\varphi,y)) \quad \text{and} \quad \forall y\ (R_i(x,y) \rightarrow \pi_r(\varphi,y)).$$

After rewriting with respect to (Def-R_i) we get for the first pattern

$$\exists y\ (\neg de_i(x) \rightarrow \exists\alpha_i\ [x\alpha_i] = y) \wedge \pi_r(\varphi,y)$$

and, as was done in the proof of Theorem 2.2.3, some simple logical equivalence preserving manipulations eliminate the equation and yield

$$\neg de_i(x) \wedge \exists\alpha_i\ \pi_r(\varphi,[x\alpha_i]).$$

Similarly for the other quantificational pattern we obtain

$$\neg de_i(x) \rightarrow \forall\alpha_i\ \pi_r(\varphi,[x\alpha_i]).$$

The process of rewriting with respect to (Def-R_i) preserves logical equivalence. Eliminating every R_i-literal from $\Pi_r(\Sigma)$ and $\Pi_r(\varphi)$ leaves us with a formula

$$\Pi_r^*(\Sigma) \wedge \neg\Pi_r^*(\varphi),$$

say, together with the definitions $\{(\text{Def-}R_i)\}_i$. Nothing will have been gained if we keep the formulae (Def-R_i) in our formula set because, in general, we cannot prevent a theorem prover from restoring R_i-literals by using the right-to-left implication of (Def-R_i). Having done the rewriting step we want to delete the axioms (Def-R_i), but deleting an axiom may turn an unsatisfiable set of formulae into a satisfiable set. In our setting, fortunately this does not happen. Proving that every model for the transformed formula set without (Def-R_i) can be turned into a model with (Def-R_i) is very easy. After all occurrences of R have been rewritten, the definitions (Def-R_i) are the only formulae remaining in which any R_i occur. This is also true, if, according to Corollary 2.4.5 more formulae Δ are added, because no R_i predicate occurs in any of the additional formulae in Δ. This means we can just use the equivalences (Def-R_i) as definitions for R_i. Therefore every model for $\Pi_r^*(\Sigma) \wedge \neg\Pi_r^*(\varphi) \wedge \Delta$, can be extended by an interpretation for the symbols R_i such that (Def-R_i) is satisfied. This is true regardless as to whether a sort AF_i is interpreted as a set of functions and $[\cdot,\cdot]$ is interpreted as the 'apply'-operation or not.

It is easy to see that the functional translation mapping Π_f is given by the sequence of operations we have just gone through: the relational translation Π_r, adding (Def-R_i), the elimination of the R_i-predicates, the elimination of the equations and deleting every (Def-R_i). Thus, Π_f can be viewed as a killer transformation.

By combining the previous results and noting that, for any model $\mathcal{M}$

$$\mathcal{M} \models \Pi_r^*(\Sigma) \wedge \neg\Pi_r^*(\varphi) \quad \text{iff} \quad \mathcal{M} \models \Pi_f(\Sigma) \wedge \neg\Pi_f(\varphi),$$

we obtain the soundness and completeness result for the functional translation (Theorem 2.4.1).

Corollary 2.4.5 is useful for making additional refining assumptions. It allows us to add sort declarations and axioms Δ to $\Pi_f(\Sigma) \wedge \neg\Pi_f(\varphi)$ while preserving satisfiability and unsatisfiability, when Δ holds in the functional extension $\mathcal{M}^*$ of a relational model $\mathcal{M}$, or there is an extension $\mathcal{M}'$ of $\mathcal{M}^*$ that makes Δ true. For example, since the functional translation generates terms of a particular form , namely $[[[[x\alpha_1]\alpha_2]\ldots]\alpha_m]$, we may consider introducing a new associative binary operation $\bullet$ defined by the sort declaration $\bullet : AF^* \times AF^* \to AF^*$ and the statement $\forall x{:}W\ \forall \alpha,\beta{:}AF^*\ [[x\alpha]\beta] = [x, \alpha\bullet\beta]$, where AF^* is a new sort. If we define $\mathcal{M}'$ as an extension of $\mathcal{M}^*$ in which AF^* is interpreted as the set of all possible compositions of functions in AF' of $\mathcal{M}'$, and $\bullet$ is interpreted as ordinary composition of functions, then, of course, $\mathcal{M}'$ satisfies the above additional definitions (collected in Δ). Then any term $[[[[x\alpha_1]\alpha_2]\ldots\alpha_m]$ is replaced by a term $[x, \alpha_1 \bullet \alpha_2 \bullet \ldots \bullet \alpha_m]$, which is what we do in the papers Ohlbach and Schmidt (1997) and Ohlbach, Schmidt and Hustadt (1996). Or, we can go further and instead of introducing $\bullet$ we can view $[\cdot,\cdot]$ as an associative operation as is done in Ohlbach (1988a). In this thesis it is important *not* to view $[\cdot,\cdot]$ as an associative operation, though I grant that my convention of omitting the brackets from $[[[[x\alpha_1]\alpha_2]\ldots]\alpha_m]$ is suggestive.

2.5 Paths and prefix stability

This section focuses on the class of first-order formulae into which any non-schema is mapped by Π_f and $\neg\Pi_f$.

In the standard semantics, the truth of a formula φ in a world x depends only on the truth of subformulae of φ in R_i-successors of x (finitely many, in $K_{(m)}$). This means for the truth of φ in x, the predecessors of x or disconnected parts of the frame are irrelevant. This is made precise in the generated model property of normal modal logics, which says: Any complete normal multi-modal logic can be characterised by the class of all generated frames for that logic. The property allows us to assume any frame is connected and has a starting world x, say.[6] This is reflected more directly in the functional setting, where worlds are denoted by 'lists' of the form

$$[x\alpha_1 \ldots \alpha_m]. \tag{2.11}$$

The list notation has a distinct advantage, as it expresses not only reachability but it displays also the *path* via $\alpha_1, \ldots, \alpha_m$ from the initial world x to that world. For this reason terms of the form (2.11) will also be referred to as *paths*, even though strictly they define worlds. (There is a natural correspondence of our reading of a path in a generated frame to the conventional reading whereby a path is a sequence of nodes in a tree or graph.)

The generated frame property of modal logics is accommodated in the functional or path encoding by the property known as the *prefix stability property* (Ohlbach 1988a) or the *unique prefix property* (Auffray and Enjalbert 1992).[7] It says, each variable in any $\Pi_f(\varphi)$ has a *unique prefix* of other variables. We will now define prefixes and prefix stability formally.

[6] By a connected frame we mean a frame, in which for any pair of worlds $(x, y) \in (R \cup R^{\smile})^*$.

[7] Auffray and Enjalbert credit Ohlbach for noting this invariance first.

The definition of prefixes extends the conventional definition of prefixes of strings. Consider a term

$$t = [x\alpha_1 \dots \alpha_i \alpha_{i+1} \dots \alpha_m].$$

in the target logic of the functional translation mapping. We say any subterm x or $[x\alpha_1 \dots \alpha_i]$ (for $1 \le i \le m$) of t is a *prefix in the term t*. Furthermore, the *prefix of a variable* α_{i+1} in the term t is the term $[x\alpha_1 \dots \alpha_i]$. The prefix of the variable α_1 is x. x has no prefix.

Let T be any set of terms of the form t as above. T is said to be *prefix stable* if any variable α of type AF occurring in T has exactly one prefix, that is, the set $\text{prefix}(\alpha, T) = \{s \mid [s\alpha \dots] \text{ occurs in } T\}$ is a singleton set. We say a *term t* is prefix stable if the singleton set $\{t\}$ is prefix stable. The *prefix of a variable α in a prefix stable set of terms* is the unique prefix of α in any term of the set.

Theorem 2.5.1 Let φ be a modal formula. Provided each α_i in the definition of π_f is a fresh variable, the set of terms occurring in $\Pi_f(\varphi)$ is prefix stable.

This can easily be seen to be the case for the example (1.6) we gave in the Preview:

$$\begin{aligned} \rho_1 &= \underset{\alpha}{\Box}(\underset{\beta}{\Diamond}\underset{\gamma}{\Box}\neg p \vee \underset{\delta}{\Diamond}\underset{\epsilon}{\Diamond} p) \quad \text{and} \\ \Pi_f(\rho_1) &= \forall x\ \forall\alpha\ (\exists\beta\forall\gamma\ \neg P[x\alpha\beta\gamma] \vee \exists\delta\exists\epsilon\ P[x\alpha\delta\epsilon]). \end{aligned}$$

Compare ρ_1 and and its translation $\Pi_f(\rho_1)$. The modal operators and the functional variables form unique pairs as indicated. α is associated with the first occurrence of a $\Box$, β with the first occurrences of a $\Diamond$, and so forth. We see the structure of ρ_1 determines a characteristic ordering on the variable occurrences in the terms $[x\alpha\beta\gamma]$ and $[x\alpha\delta\epsilon]$ of $\neg\Pi_f(\rho_1)$. This ordering of the variables is formalised by prefix stability.

Any term (or path) in $\neg\Pi_f(\varphi)$ is a list of one world variable and a sequence of functional variables and the world variable is always the same, as in this example.

$$\neg\Pi_f(\rho_1) = \exists x\ \exists\alpha\ (\forall\beta\exists\gamma\ P[x\alpha\beta\gamma] \wedge \forall\delta\forall\epsilon\ \neg P[x\alpha\delta\epsilon])$$

We do away with the world quantifier $\exists x$ by introducing a special world constant $[]$, the empty list, which we view as the initial world. So, in future we regard $\neg\Pi_f(\varphi)$ to coincide with $\neg\pi_f(\varphi, [])$. Thus, terms in $\neg\Pi_f(\varphi)$ will have the form

$$[\alpha_1 \dots \alpha_m],$$

as opposed to $[x\alpha_1 \dots \alpha_m]$. The reformulation of $\neg\Pi_f(\rho_1)$ is

$$\exists\alpha\ (\forall\beta\exists\gamma\ P[\alpha\beta\gamma] \wedge \forall\delta\forall\epsilon\ \neg P[\alpha\delta\epsilon]). \tag{2.12}$$

Evidently, Theorem 2.5.1 holds also for $\neg\Pi_f(\varphi)$.

Because prefix stability of terms is a fundamental concept, it is instructive to give an independent definition of target logic of $\neg\Pi_f$. This definition emphasises the particular ordering of the variables in any literal $\pm P[\dots]$ as determined by prefix stability (and the original modal formula). For reasons that will become clear in the next chapter (where we define path logics that arise by the optimised functional translation), we call the logic *basic* *non-optimised path logic*. The basic non-optimised path logic is associated with the logics K, KD and their multi-modal versions for which the theory $\Pi_f(\Sigma)$ is empty.

The language of basic non-optimised path logic is that of monadic sorted first-order logic extended with a non-associative binary operation $[\cdot,\cdot]$ and a designated constant $[]$. The sorts are W, $\{AF_i\}_i$ and AF^*. There are finitely many unary predicate symbols $P, Q, \ldots$ and possibly also special unary predicates $\{de_i\}_i$. Functional variables are denoted by Greek letters $\alpha, \beta, \ldots$ and are possibly indexed by functional subsorts. The constant $[]$ has sort world. The function $[\cdot,\cdot]$ maps pairs of world terms and functional terms to world terms. Terms, also called *paths*, are of the form

$$[[[[[]\alpha_1]\alpha_2]\ldots]\alpha_m] \quad \text{or in shorthand notation} \quad [\alpha_1\alpha_2\ldots\alpha_m].$$

In the language of basic non-optimised path logic there are no world variables and no compound functional terms.

Let V be a finite set of unary predicate symbols and let $X_m = \{\alpha_1, \ldots, \alpha_m\}$ be an ordered set of variables. An *atomic basic non-optimised path formula* of V over X_i is a monadic literal with an argument of length i:

$$P[\alpha_1\ldots\alpha_i] \quad \text{or} \quad \neg P[\alpha_1,\ldots,\alpha_i]$$

Basic non-optimised path formulae are defined inductively by:

(i) Any atomic basic non-optimised path formula over X_i is a basic non-optimised path formula over X_i.

(ii) $\exists\alpha_{i+1}\,\varphi$ and $\forall\alpha_{i+1}\,\varphi$ are basic non-optimised path formulae over X_i, when φ is a basic non-optimised path formula over X_{i+1}.

(iii) Any Boolean combination of basic non-optimised path formulae over X_i is a basic non-optimised path formula over X_i. For example, $\varphi \to \psi$, $\neg\varphi$, $\varphi \wedge \psi$, etcetera, are basic non-optimised path formulae over X_i, when both φ and ψ are.

A sample basic non-optimised path formula over X_3 is the formula

$$\exists\alpha_1\ (\forall\alpha_2\exists\alpha_3\ P[\alpha_1\alpha_2\alpha_3] \wedge \forall\alpha_2\forall\alpha_3\ \neg P[\alpha_1\alpha_2\alpha_3]).$$

Note, it a reformulation in terms of a minimal number of variables of the formula in (2.12) above.

Theorem 2.5.2 Let φ be a basic non-optimised path formula. The set of terms occurring in φ is prefix stable.

Proof. Not difficult. □

Theorem 2.5.3 Let φ be any modal formula. $\neg\Pi_f(\varphi)$ is equivalent to a basic non-optimised path formula.

Proof. Let m be the modal degree of φ. Associate with each variable in $\neg\pi_f(\varphi, [])$ a variable in X_m in an economical way. □

Theorem 2.5.4 Let ψ be any basic non-optimised path formula. Then a modal formula φ exists such that ψ is equivalent to $\neg\Pi_f(\varphi)$.

Proof. Define a mapping * from basic non-optimised path formulae to modal formulae inductively by

$$\begin{aligned}(P_i[\alpha_1 \dots \alpha_m])^* &= p_i \\ (\bot)^* &= \bot^* \\ (\phi_1 \to \phi_2)^* &= \phi_1^* \to \phi_2^* \\ (\exists \alpha_i\ \phi)^* &= \Diamond \phi^*\end{aligned}$$

in such a way that p_i is a propositional variable uniquely associated with the predicate symbol P_i. Now, let $\varphi = \neg\psi^*$. φ is a well-defined modal formula and $\neg\pi_f(\varphi, []) = \psi$, provided the variables are chosen appropriately from X_m. □

The two theorems establish a one-one correspondence between KD-formulae and basic non-optimised path formulae. Even though we do not define a formal calculus it is immediate that basic non-optimised path logic is a decidable first-order class, since KD is decidable.

We digress in order to establish connections to related decidable fragments of first-order logic. There is a one-to-one correspondence between basic non-optimised path logic and the class of *ordered first-order formulae* that Herzig defines in (1990). The correspondence is determined by mapping every atom of the form $P[\alpha_1 \dots \alpha_m]$ to the m-ary atom of the form $P(\alpha_1, \dots, \alpha_m)$. By the same mapping basic non-optimised path logic can been seen to be a proper fragment of a decidable fragment of first-order logic, called fluted logic, that is part of Quine's predicate functor logic. In fact, the definition of basic non-optimised path logic was inspired by the definition of fluted logic found in Purdy (1996a, 1996b). The mapping is also useful for encoding modal formulae into first-order logic without function symbols for which the performance of theorem provers is better. For details refer to Hustadt and Schmidt (1997a, 1997b), where we describe also how multi-modal formulae can be converted to unsorted first-order formulae.

The concept of prefix stability as defined above, does not apply naturally to clauses, the problem being that due to Skolemisation terms are no longer lists of variables. The definition of prefix stability can be adapted as proposed in Ohlbach (1991) or Auffray and Enjalbert (1992) by distinguishing different occurrences of a variable (inside or outside the scope of a Skolem function). This is not enough. To achieve preservation of the adapted form of prefix stability an *extended* (or *strong*) form of Skolemisation as proposed by Herzig (1989) needs to be applied. Zamov (1989) proposes a similar solution that involves the notion of *tree-likeness*. We will not have to adopt any of these solutions as the optimised functional translation eliminates from the clausal forms all complex Skolem terms except for constants.

2.6 Translating non-serial into serial modal logics

This section establishes a connection between non-serial and serial modal logics that allows us to focus for much of our exposition on serial modal logics. The modal logic K can be translated into the logic KD adjoined with a special *propositional* variable de. $KD + \{de\}$ is a conservative extension of K.[8]

[8] I suspect this result is known, though I have not come across it in the literature.

The translation * from K to KD is defined inductively by:

$$\begin{aligned} p^* &= p \\ (\bot)^* &= \bot \\ (\varphi \to \psi)^* &= \varphi^* \to \psi^* \\ (\Diamond\varphi)^* &= \neg de \wedge \Diamond\varphi^* \end{aligned}$$

Informally, the new propositional variable de has the same interpretation as the dead-end predicate of the functional translation. It denotes the set of worlds at which the accessibility relation of K is not defined.

Lemma 2.6.1 Let $\mathcal{M} = (W, R, \iota)$ be any K-model. Let R' be a total extension of R and let De be a subset of W such that

$$(x, y) \in R \quad \text{iff} \quad x \notin \text{De and } (x, y) \in R'.$$

Let $\mathcal{M}' = (W, R', \iota')$ be a KD-model such that for all atomic p except for de, $\iota(p) = \iota'(p)$, and $\iota'(de) = \text{De}$. Then for all K-formulae φ and any $x \in W$,

$$\mathcal{M}, x \models \varphi \quad \text{iff} \quad \mathcal{M}', x \models \varphi^*.$$

Proof. By induction on the structure of φ. The base case is easy: $\iota(p) = \iota'(p) = \iota'(p^*)$. Suppose for any $x \in W$, $\mathcal{M}, x \models \varphi$ iff $\mathcal{M}', x \models \varphi^*$. the inductive hypothesis. The Boolean cases are easy. For example, $\mathcal{M}, x \models \neg\varphi$ iff $\mathcal{M}, x \not\models \varphi$ iff $\mathcal{M}', x \not\models \varphi^*$ iff $\mathcal{M}', x \models \neg\varphi^*$. We consider the modal case. $\mathcal{M}', x \models (\Diamond\varphi)^*$ iff $\mathcal{M}', x \models \neg de \wedge \Diamond\varphi^*$, iff $\mathcal{M}', x \not\models de$ and $\mathcal{M}', x \models \Diamond\varphi^*$, iff $x \notin \text{De}$ and $\exists y\ (x, y) \in R' \wedge \mathcal{M}', x \models \varphi^*$, iff $x \notin \text{De}$ and $\exists y\ (x, y) \in R' \wedge \mathcal{M}, x \models \varphi^*$, iff $\exists y\ (x, y) \in R \wedge \mathcal{M}, x \models \varphi^*$, iff $\mathcal{M}, x \models \Diamond\varphi^*$. $\square$

Theorem 2.6.2 Any formula φ is provable in K iff φ^* is provable in KD.

Proof. ($\Rightarrow$) Any proof of φ in K translates to a proof of φ^* in KD, if the translations of the axioms and rules of K are provable in KD. The translation of the K-axiom is:

$$\begin{aligned} (\Box(p \to q) \to (\Box p \to \Box q))^* &\equiv (\neg de \to \Box(p \to q)) \to ((\neg de \to \Box p) \to (\neg de \to \Box q)) \\ &\equiv (\neg de \to \Box(p \to q)) \to (\neg de \to (\Box p \to \Box q)) \\ &\equiv \neg de \to (\Box(p \to q)) \to (\Box p \to \Box q)) \end{aligned}$$

which is provable in KD by implication introduction. Showing that modus ponens and necessitation for K are provable in KD is straightforward. For instance, assume φ^* is provable in KD. Then $\Box\varphi^*$ is provable in KD and so is $\neg de \to \Box\varphi^*$ which is the same as $(\Box\varphi)^*$.

($\Leftarrow$) We prove the contrapositive. Suppose φ is not provable in K. Then since K is complete, $\not\models_K \varphi$, that is, there is a K-model $\mathcal{M} = (W, R, \iota)$ that falsifies φ. Define $\mathcal{M}' = (W, R', \iota')$ as in the previous lemma. By that result $\mathcal{M}, x \models \varphi$ iff $\mathcal{M}', x \models \varphi^*$. Thus, $\mathcal{M}'$ does not satisfy φ^*, that is, $\not\models_{KD} \varphi^*$. Since KD is complete, it follows φ^* is not provable in KD. $\square$

This result generalises to extensions of K with any theory:

Any formula φ is provable in $K\Sigma$ iff φ^* is provable in $KD\Sigma^*$.

For the ($\Rightarrow$) direction we need to convince ourselves that the translation of each axiom in Σ is derivable from Σ^*, but this is clear. The argument for the converse direction does not require change.

For example, *KD45* is then defined by $KD + \{de\}$ extended by the axioms

$$\begin{array}{ll} 4^* & (\neg de \to \Box p) \to (\neg de \to \Box(\neg de \to \Box p)) \\ 5^* & (\neg de \wedge \Diamond p) \to (\neg de \to \Box(\neg de \wedge \Diamond p)), \end{array}$$

which simplify to $\Box p \to (\neg de \to \Box(\neg de \to \Box p))$ and $(\neg de \wedge \Diamond p) \to \Box(\neg de \wedge \Diamond p)$, respectively.

The generalisation to the multi-modal setting is obtained similarly. By adjoining propositional variables de_i to a given multi-modal logic, we can encode any non-serial modality $\Diamond_i$ in terms of a serial modality.

Corollary 2.6.3 Any formula φ is provable in $K_{(m)}\Sigma$ iff φ^* is provable in its serial extension.

Chapter 3

The optimised functional translation

This chapter introduces the optimisation operator Υ. It extends the applicability of first-order theorem proving methods to modal logics that are ordinarily not first-order definable and transforms modal formulae to particularly simple clausal form for which in many cases satisfiability can be decided by standard resolution. Important for both properties, 'extended first-order definability' and decidability, is that Υ swaps existential and universal functional quantifiers.

The idea for swapping quantifiers and simplifying the translation of non-schemas (of the formulae we which to prove are theorems) was first mentioned by Herzig and others (Fariñas del Cerro and Herzig 1988, Herzig 1989) and it features in Ohlbach (1988a). This chapter presents the core results announced in Ohlbach (1993a) and proved in Ohlbach and Schmidt (1997). We show the optimisation applies not only to non-schemas, it applies also to schemas, bearing first-order functional correspondences and a first-order calculus for essentially second-order modal logics. The application of the quantifier exchange operator Υ hinges on modal logics being determined by maximal functional models. This is established in Section 3.1. Section 3.2 defines the operator Υ and proves the necessary completeness theorems for the translation by $\Upsilon\Pi_f$. Section 3.3 is concerned with correspondence theory exemplified for the schema M.

3.1 Exchanging quantifiers in maximal functional models

We will show that in maximal functional models enough functions are available so that the following equivalence is true.

$$\exists\alpha\forall\beta\ \psi \leftrightarrow \forall\beta\exists\alpha\ \psi \tag{3.1}$$

Swapping quantifiers does not work in the relational setting. Suppose a formula $\Box\Diamond p$ is true at the world x of the model depicted in Figure 3.1. For every world y accessible from x there is a world accessible from y where p is true, that is,

$$\forall y\ (R(x,y) \rightarrow \exists z\ (R(y,z) \wedge \text{ `}p \text{ is true in } z\text{'})).$$

Suppose the situation is as in Figure 3.1. In this model we have swapped the existential

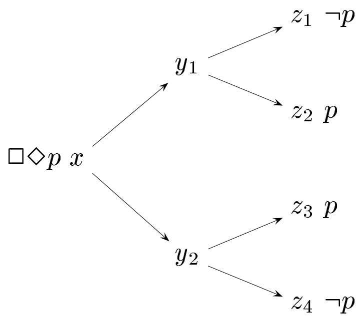


Figure 3.1: A relational model $\mathcal{M}$

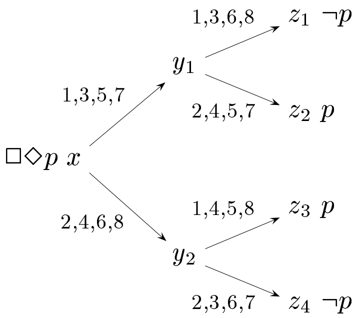


Figure 3.2: The maximal functional extension $\mathcal{M}^m$

quantifier $\exists z$ with the universal quantifier $\forall y$ and evidently the formula

$$\exists z \forall y\ (R(x,y) \rightarrow (R(y,z) \wedge \text{ `}p \text{ is true in } z\text{'}))$$

is false.

But now consider the maximal functional model defined by Figure 3.2 (that combines eight frames including the four frames of Figure 2.3 of the previous Chapter). The labels i abbreviate the corresponding accessibility functions α_i. In the functional language we can express the fact that $\Box\Diamond p$ is true at x by

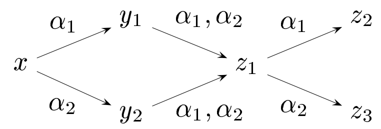
$$\forall\alpha\exists\beta \text{ `}p \text{ is true at } [x\alpha\beta]\text{'}.$$

In the depicted maximal model we can swap the $\exists\beta$ and the $\forall\alpha$ quantifiers:

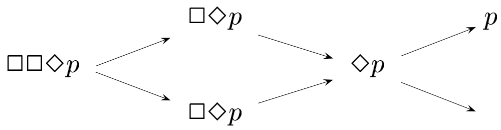
$$\exists\beta\forall\alpha \text{ `}p \text{ is true at } [x\alpha\beta]\text{'}$$

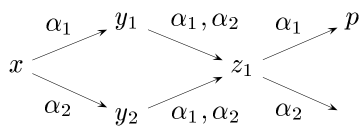
is also true at x, because the function α_4 (as well as the function α_5) maps the worlds y_1 and y_2 to a world where p holds. Moreover, regardless in which one of the worlds z_1, z_2, z_3 or z_4, p is true, in this model there is always a function α_i which maps y_1 and y_2 to the right worlds. We will show that in any functional frames which is *maximal*, it is always justified to move the existential quantifiers in front of the universal quantifiers. The previous chapter proves that a relational frame can always be associated with a unique maximal functional frame, and this is what we need.

For tree-like structures the decomposition into function sets is easier than for arbitrary structures. Consider the frame



and suppose $\Box\Box\Diamond p$ is true at x. For all worlds y accessible from x (y_1 and y_2) and all worlds z accessible from y (z_1) there is a world z' accessible from z such that p is true at z'. The existentially quantified z' depends on y and z. In this frame there is only one possible instance for z (namely z_1). Therefore, z' depends actually only on y. The dependencies can be as follows: for $y = y_1$ assign z_2 to z' and for $y = y_2$ assign z_3 to z' (or vice versa). Different paths for reaching $z = z_1$ may continue to different worlds. Fortunately in propositional modal logics this situation does not arise, as truth does not depend on predecessor worlds (in quantified modal logic it can). If one path crossing z_1 is extended to a world satisfying some p then all other paths crossing z_1 can be extended to the same world. To illustrate this, consider again the formula $\Box\Box\Diamond p$ which is true at x. Then $\Box\Diamond p$ is true at y_1 and y_2 and $\Diamond p$ is true at z_1. It is sufficient that p is true at either z_2 or z_3, not necessarily in both worlds.





The left picture is a model for the formula $\Box\Box\Diamond p$. In the corresponding functional model (depicted on the right)

$$\forall \beta\beta' \exists \gamma \text{ `}p \text{ is true at } [x\beta\beta'\gamma]\text{'}$$

is true. If we let β be α_1 then we can move the $\exists\gamma$ quantifier in front of $\forall\beta\beta'$. In this model

$$\exists\gamma \forall \beta\beta' \text{ `}p \text{ is true at } [x\beta\beta'\gamma]\text{'}$$

is satisfied. We have exploited the fact that in propositional modal logics, truth of a formula in a particular world does not depend on predecessor worlds.

The next theorem proves the equivalence (3.1) necessary for swapping functional quantifiers. It requires the following lemma. We formulate the results for serial models defined by one binary relation.

Lemma 3.1.1 Let ψ be a formula of a non-optimised path logic. Let α be a variable in ψ with prefix $[xs]$. Let $\mathcal{M} = (W, AF, [\cdot,\cdot], \iota)$ be a functional model in which $[xs] = \underline{y}$, and $\underline{\alpha}_1$ and $\underline{\alpha}_2$ are two functions in AF such that $[\underline{y}\underline{\alpha}_1] = [\underline{y}\underline{\alpha}_2]$.[1] Let $\mathcal{M}[\alpha/\underline{\alpha}_1]$ and $\mathcal{M}[\alpha/\underline{\alpha}_2]$ be two models that are like $\mathcal{M}$ except that α is assigned the value $\underline{\alpha}_1$ and $\underline{\alpha}_2$, respectively. Then for any $x \in W$

$$\mathcal{M}[\alpha/\underline{\alpha}_1], x \models \psi \quad \text{iff} \quad \mathcal{M}[\alpha/\underline{\alpha}_2], x \models \psi.$$

Proof. Without loss of generality we assume ψ is in prenex normal form, that is, ψ consists of a quantifier prefix followed by a quantifier-free matrix ψ'. Any term in which α occurs has the form $[xs\alpha\ldots]$. Suppose the variables in s are $\gamma_1, \ldots, \gamma_n$. The γ_i may be existentially

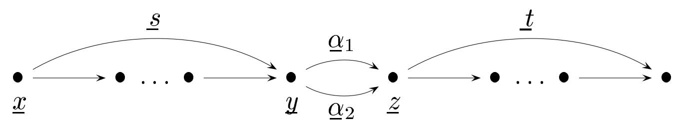
[1] Underlined symbols refer to concrete elements in the models.

or universally quantified. Define two interpretations $\mathcal{M}_1$ and $\mathcal{M}_2$ which are like $\mathcal{M}$ except that in $\mathcal{M}_1$, α has the value $\underline{\alpha}_1$, and in $\mathcal{M}_2$, α has the value $\underline{\alpha}_2$. That is,

$$\mathcal{M}_1 = \mathcal{M}[x/\underline{x}, \gamma_1/\underline{\gamma}_1, \dots, \gamma_n/\underline{\gamma}_n, \alpha/\underline{\alpha}_1] \quad \text{and} \quad \mathcal{M}_2 = \mathcal{M}[x/\underline{x}, \gamma_1/\underline{\gamma}_1, \dots, \gamma_n/\underline{\gamma}_n, \alpha/\underline{\alpha}_2].$$

By induction on the structure of ψ' we show that $\mathcal{M}_1$ satisfies ψ' iff $\mathcal{M}_2$ satisfies ψ'.

Only the base case with $\psi' = P[xs\alpha t]$ is non-trivial. By prefix stability, α does not occur in s and it does not occur in t. Thus, the interpretation of s and t does not depend on the assignment for α. Let the interpretations of x, s and the suffix t in $\mathcal{M}_1$ and $\mathcal{M}_2$ be the same, say $\underline{x}$, $\underline{s}$ and $\underline{t}$, respectively. $\underline{s}$ and $\underline{t}$ and are compositions of functions in AF. Suppose $[\underline{y}\underline{\alpha}_1] = \underline{z} = [\underline{y}\underline{\alpha}_2]$. We have the following situation.



The value of $[xs\alpha t]$ in $\mathcal{M}_1$ is $[\underline{x}\underline{s}\underline{\alpha}_1\underline{t}]$ and $[\underline{x}\underline{s}\underline{\alpha}_1\underline{t}] = [\underline{y}\underline{\alpha}_2\underline{t}] = [\underline{x}\underline{s}\underline{\alpha}_2\underline{t}]$ which is the value of $[xs\alpha t]$ in $\mathcal{M}_2$. It follows that the value for $[xs\alpha t]$ in $\mathcal{M}_1$ is in $\iota_1(P)$ iff the value for $[xs\alpha t]$ in $\mathcal{M}_2$ is in $\iota_2(P)$, since $\iota_1(P) = \iota_2(P) = \iota(P)$. ($\iota_1$ and ι_2 are the valuation mappings of $\mathcal{M}_1$ and $\mathcal{M}_2$.) This proves the base case.

The induction step is a straightforward application of the induction hypothesis. We omit the details and conclude, $\mathcal{M}_1$ satisfies ψ' iff $\mathcal{M}_2$ does too. There are no conditions on the assignments to the variables γ_i in s. Thus, if a γ_i is universally quantified, we take all assignments of accessibility functions to γ_i, and if γ_i is existentially quantified, we choose an appropriate assignment. Consequently, $\mathcal{M}[\alpha/\underline{\alpha}_1] \models \psi$ iff $\mathcal{M}[\alpha/\underline{\alpha}_2] \models \psi$. □

Recall from Section 2.4 the definition of the functional extension of a relational model. The functional extension of a serial relational model $\mathcal{M} = (W, R, \iota)$ is the model $\mathcal{M}^* = (W, R, AF_R^m, [\cdot,\cdot], \iota)$. By definition, AF_R^m is the largest set of all functions that define R.

In the next theorem, we will write $\psi[xs\dots]$ for $\psi[[xs\dots]]$, meaning the first-order formula ψ with one or more occurrences of the term $[xs\dots]$.

Theorem 3.1.2 Let φ be of a modal formula true in a relational model $\mathcal{M}$. Let $\mathcal{M}^*$ be its functional extension. For the functional translation of φ the following equivalence is true in $\mathcal{M}^*$:

$$\forall\alpha\exists\beta\ \psi[xs\alpha t\beta t'] \leftrightarrow \exists\beta\forall\alpha\ \psi[xs\alpha t\beta t']. \tag{3.2}$$

Proof. The right-to-left direction is a valid predicate logic implication. For the left-to-right direction, suppose the left hand side of the equivalence is true in $\mathcal{M}^*$. Define an arbitrary interpretation $\mathcal{M}^{*\prime}$ which is like $\mathcal{M}^*$ for x, s and the predicate symbols and satisfies $\forall\alpha\exists\beta\ \psi[xs\alpha t\beta t']$. That is $\mathcal{M}^{*\prime} = \mathcal{M}^*[\vec{P}/\underline{\vec{P}}, x/\underline{x}, \vec{\gamma}/\underline{\vec{\gamma}}]$ is such that

$$\forall\alpha\exists\beta\ \psi[xs\alpha t\beta t'] \text{ is true in } \mathcal{M}^{*\prime}.$$

Let ψ' denote the formula ψ with every occurrence of the term $[xs\alpha t\beta t']$ replaced by t'. For any $\underline{y} \in W$ define $B(\underline{y})$ by

$$\{\underline{\beta} \in AF_R \mid \underline{y} \text{ is the value of } [xs\alpha t] \text{ and } \psi' \text{ is true at } [\underline{y}\underline{\beta}] \text{ in } \mathcal{M}^{*\prime}\}.$$

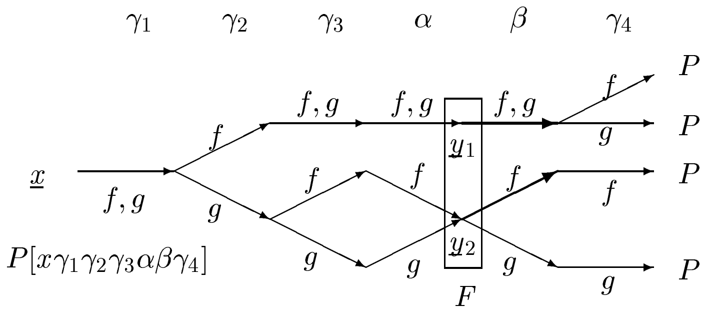
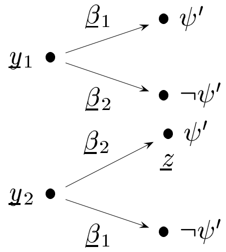


Figure 3.3: A model of $\exists P\ \exists x\ \exists\gamma_1\forall\gamma_2\gamma_3\alpha\exists\beta\forall\gamma_4\ P[x\gamma_1\gamma_2\gamma_3\alpha\beta\gamma_4]$

$B(\underline{y})$ is the set of functions in $\mathcal{M}^{*\prime}$ that map the world $\underline{y}$ to worlds in which ψ' is true. Let F be the set of all $B(\underline{y})$. We want to exhibit for $\mathcal{M}^{*\prime}$ the existence of a function $\underline{\beta}_0$ in AF_R such that for every world $\underline{y}$ a function $\underline{\beta}$ exists in $B(\underline{y})$ with $[\underline{y}\underline{\beta}_0] = [\underline{y}\underline{\beta}]$. It suffices to show

$$\forall \underline{y}_1\underline{y}_2\ B(\underline{y}_1) \cap B(\underline{y}_2) \neq \emptyset.$$

Suppose not. Suppose two worlds $\underline{y}_1$ and $\underline{y}_2$ exists with $B(\underline{y}_1)$ and $B(\underline{y}_2)$ disjoint. Let $\underline{\beta}_1$ be any function in $B(\underline{y}_1)$ and $\underline{\beta}_2$ any function in $B(\underline{y}_2)$. The situation is as follows:



Consider the function $\underline{\beta}$ which is like $\underline{\beta}_1$ except that $[\underline{y}_2\underline{\beta}] = \underline{z}$, that is, $\underline{\beta}$ maps $\underline{y}_2$ not to a world where ψ' is false but to $\underline{z}$ where ψ' is true. Since AF_R contains all total functions from W to W, $\underline{\beta}$ is in AF_R. But then $\underline{\beta} \in B(\underline{y}_1) \cap B(\underline{y}_2)$. Therefore $B(\underline{y}_1)$ and $B(\underline{y}_2)$ cannot be disjoint.

For every interpretation $\mathcal{M}^{*\prime}$ we can find some $B(\underline{y}) \in F$ and some $\underline{\beta} \in B(\underline{y})$, for $\underline{y}$ the value of $[xs\alpha t]$, ψ is true in $\mathcal{M}^{*\prime}[\beta/\underline{\beta}]$ and $[\underline{y}\underline{\beta}_0] = [\underline{y}\underline{\beta}]$. Now, we can apply Lemma 3.1.1 and get

$$\mathcal{M}^{*\prime}[\beta/\underline{\beta}] \models \psi \quad \text{iff} \quad \mathcal{M}^{*\prime}[\beta/\underline{\beta}_0] \models \psi.$$

Because $\mathcal{M}^{*\prime} \models \forall\alpha\exists\beta\ \psi$ and because the assignment to α is arbitrary we can conclude that $\mathcal{M}^{*\prime}[\beta/\underline{\beta}_0] \models \forall\alpha\ \psi$. Hence $\mathcal{M}^{*\prime} \models \exists\beta\forall\alpha\ \psi$. $\square$

The following example illustrates the notions of the proof. Figure 3.3 is a model for the formula

$$\exists P\ \exists x\ \exists\gamma_1\forall\gamma_2\gamma_3\alpha\exists\beta\forall\gamma_4\ P[x\gamma_1\gamma_2\gamma_3\alpha\beta\gamma_4].$$

Here $\{f, g\} \subseteq B(\underline{y}_1)$ and $\{f, g\} \subseteq B(\underline{y}_2)$. Choosing $\underline{\beta}_0 = f$ (highlighted by the thick arrows) as assignment for β makes this model satisfy

$$\exists P\ \exists x\ \exists\gamma_1\forall\gamma_2\gamma_3\exists\beta\forall\alpha\gamma_4\ P[x\gamma_1\gamma_2\gamma_3\alpha\beta\gamma_4].$$

AF_R being the maximal set of functions that determine the defining frames, it contains more functions than merely f and g.

The generalisation of the lemma and the theorem to non-serial and multi-modal models is routine.

Since $\varphi \leftrightarrow \psi$ and $\neg\varphi \leftrightarrow \neg\psi$ are logically equivalent, the equivalence (3.2) remains true if both the left-hand side and the right-hand side are negated. This implies that in the functional translation of modal formulae, as well as in their negation, the quantifiers may be exchanged, when the sorts AF_i are interpreted as a *maximal* set of accessibility functions.

3.2 The quantifier exchange operator

We exploit the equivalence (3.1) for moving existential quantifiers *outward* in the negated form of the functional translation of non-schemas and for moving existential quantifiers *inward* in the translation of schemas.

Formally, the so-called *quantifier exchange* operator Υ converts a non-optimised path formula into prenex normal form and moves *all* existential quantifiers of functional variables inwards as far as possible according to the rule

$$\exists\alpha\forall\beta\ \psi \quad \text{becomes} \quad \forall\beta\exists\alpha\ \psi.$$

Now, for any modal formula φ, $\Upsilon\Pi_f(\varphi)$ has a quantifier prefix consisting of a universally quantified world variable followed by a sequence of universally quantified variables of sort AF_i and a sequence of existentially quantified variables of sort AF_i. The quantifier prefix of the negation $\neg\Upsilon\Pi_f(\varphi)$ is then a sequence of existential quantifiers (including the one existential world quantifier) followed by a sequence of universal quantifiers.

Here is an example (in addition to the example given in the Preview). Consider McKinsey's schema $\Box\Diamond p \to \Diamond\Box p$. Since D is a theorem in KM, the functional translation $\Pi_f(M)$ is given by

$$\forall P\ \forall x\ (\forall\alpha\exists\beta\ P[x\alpha\beta]) \to (\exists\alpha'\forall\beta'\ P[x\alpha'\beta']).$$

The prefix normal form is

$$\forall P\ \forall x\ \exists\alpha\forall\beta\exists\alpha'\forall\beta'\ P[x\alpha\beta] \to P[x\alpha'\beta'].$$

Applying Υ yields

$$\forall P\ \forall x\ \forall\beta\beta'\exists\alpha\alpha'\ P[x\alpha\beta] \to P[x\alpha'\beta'].$$

The negation $\neg\Upsilon\Pi_f(M)$ is given by

$$\exists P\ \exists x\ \exists\beta\beta'\forall\alpha\alpha'\ P[x\alpha\beta] \wedge \neg P[x\alpha'\beta']. \tag{3.3}$$

and in clausal form (with Skolem constants $\underline{\beta}, \underline{\beta}'$ and $[]$, for $\underline{x}$):

1. $P[\alpha\underline{\beta}]$
2. $\neg P[\alpha'\underline{\beta}']$.

The example demonstrates the operation Υ moves functional existential quantifiers inward over universal quantifiers. This causes the opposite movement in the transformation by $\neg\Upsilon\Pi_f$ of non-schemas. In $\neg\Upsilon\Pi_f(\varphi)$ all existential quantifiers precede all universal quantifiers so that existential quantifiers generate just Skolem constants, no complex Skolem terms. (Chapters 6 onwards will be concerned with classes of such clauses.)

Applying Υ to a formula ψ results in a weaker formula ψ', since $\psi \to \psi'$, for in general, $\exists x \forall y\ \psi''(x, y)$ logically implies $\forall y \exists x\ \psi''(x, y)$, but not conversely. For any first-order formula ψ the following is logically valid:

$$\psi \to \Upsilon(\psi) \quad \text{and} \quad \neg\Upsilon(\psi) \to \neg\psi.$$

Refuting $\neg\Upsilon(\psi)$ instead of $\neg\psi$ therefore means we are in fact proving a weaker version of the formula than we originally wanted. The next result provides the conditions under which working with the weaker form does suffice for proving ψ.

Corollary 3.2.1 Let *K(D)*Σ be a complete (respectively complete and first-order definable) modal logic. For any modal formula φ, φ is a theorem in *K(D)*Σ iff

$$\Pi_f(\Sigma) \to \Upsilon\Pi_f(\varphi) \text{ is a second-order (resp. first-order) theorem.}$$

Proof. By Theorem 2.4.1, φ is a *K(D)*Σ-theorem iff $\Pi_f(\Sigma) \wedge \neg\Pi_f(\varphi)$ is unsatisfiable. We prove

$$\Pi_f(\Sigma) \wedge \neg\Pi_f(\varphi) \text{ is satisfiable} \quad \text{iff} \quad \Pi_f(\Sigma) \wedge \neg\Upsilon\Pi_f(\varphi) \text{ is satisfiable.}$$

If $\Pi_f(\Sigma) \wedge \neg\Pi_f(\varphi)$ is satisfiable, by Theorem 3.1.2, it has a functional model, in which $\neg\Pi_f(\varphi)$ and $\neg\Upsilon\Pi_f(\varphi)$ are equivalent. Hence, $\Pi_f(\Sigma) \wedge \neg\Upsilon\Pi_f(\varphi)$ is true in the model as well.

Now suppose $\Pi_f(\Sigma) \wedge \neg\Upsilon\Pi_f(\varphi)$ has a model. Υ moves existential quantifiers to the inside. Thus classically, $\Pi_f(\varphi) \to \Upsilon\Pi_f(\varphi)$ and contrapositively $\neg\Upsilon\Pi_f(\varphi) \to \neg\Pi_f(\varphi)$. Therefore, $\Pi_f(\Sigma) \wedge \neg\Pi_f(\varphi)$ is true in the same model. $\square$

The question now is: Can we go further? Can we use the quantifier exchange operation also for the translations of modal *axiom schemas*? One way of reducing the translation of any modal schema like McKinsey's schema to first-order logic is by exchanging existential and universal quantifiers. Unfortunately, this weakens the schema. This is certainly sound: Any theorem that can be proved using weaker schemas also follows from the original schemas. However, weakening a schema may be a source for incompleteness. We may not be able to prove all theorems in the weaker system.

There are two possibilities. First, we may try exploiting Theorem 3.1.2 and its Corollary which say that in certain functional models the quantifier exchange rule preserves equivalence. In the proof we needed the maximality condition of functional models. Unfortunately, this condition is not first-order definable. An infinite approximation is:

$$\begin{aligned}
\mu = {} & \forall x_1 x_2\ (x_1 \neq x_2 \to \\
& \qquad \forall x\ (\forall \alpha_1, \alpha_2 \exists \beta\ [x\alpha_1] = [x\beta] \wedge [x\alpha_2] = [x\beta])) \\
& \wedge \forall x_1 x_2 x_3\ (x_1 \neq x_2 \wedge x_1 \neq x_3 \wedge x_2 \neq x_3 \to \\
& \qquad \forall x\ (\forall \alpha_1, \alpha_2, \alpha_3\ \exists \beta\ [x\alpha_2] = [x\beta] \wedge [x\alpha_3] = [x\beta])) \\
& \wedge \forall x_1 x_2 x_3 x_4\ \ldots \\
& \vdots
\end{aligned}$$

μ is true in maximal functional models. By Corollary 2.4.5, we can add μ to $\Pi_f(\Sigma) \wedge \neg\Pi_f(\varphi)$ without changing the consistency. Under the assumption μ, $\Pi_f(\Sigma)$ and $\Upsilon\Pi_f(\Sigma)$ are equivalent, and we are licensed to make use of the quantifier exchange operation. But this means, that we may actually need to use μ in the process of finding a refutation. Since μ is infinite, this is not practical.

Another possibility for ensuring completeness is the following. If we prove, instead of $\Pi_f(\varphi)$, the *weaker* theorem $\Upsilon\Pi_f(\varphi)$, it may turn out that the weaker schemas are sufficient to prove the weaker theorems, without assuming μ. The next corollary confirms this, thus licensing for modal schemas that quantifiers may be swapped.

Theorem 3.2.2 Let *K(D)*Σ be any complete (respectively complete and first-order definable) modal logic with modus ponens and the necessitation being the only rules. Then, for any modal formula φ,

$$\varphi \text{ is a theorem in } K(D)\Sigma \quad \text{iff} \quad \Upsilon\Pi_f(\Sigma) \to \Upsilon\Pi_f(\varphi) \text{ is a s.o. (resp. f.o.) theorem}$$

Proof. For the ($\Leftarrow$) direction, suppose $\Upsilon\Pi_f(\Sigma) \to \Upsilon\Pi_f(\varphi)$ is a theorem. Since $\Pi_f(\Sigma) \to \Upsilon\Pi_f(\Sigma)$ holds, $\Pi_f(\Sigma) \to \Upsilon\Pi_f(\varphi)$ holds. This implies $\Pi_f(\Sigma) \wedge \neg\Upsilon(\Pi_f(\varphi))$ is unsatisfiable. Therefore, by Corollary 3.2.1, φ is a theorem of *K(D)*Σ.

For the ($\Rightarrow$) direction, suppose φ is a theorem of *K(D)*Σ. Then φ is either an instance of a schema or it can be obtained by repeatedly applying the rules of *K(D)*Σ. The desired result follows by induction on the length of the proof sequence if we can show, (i) for all instances ψ of schemas in Σ,

$$\Upsilon\Pi_f(\Sigma) \to \Upsilon\Pi_f(\psi),$$

and (ii) for all applications of rules 'from φ_1 and ... and φ_n infer φ' in Σ,

$$\textstyle\bigwedge_i(\Upsilon\Pi_f(\Sigma) \to \Upsilon\Pi_f(\varphi_i)) \to (\Upsilon\Pi_f(\Sigma) \to \Upsilon\Pi_f(\varphi)).$$

Let $\psi[\vec{p}/\underline{\vec{p}}]$ be an instance of a schema. We have

$$\begin{aligned} \Upsilon\Pi_f(\Sigma) \to \Upsilon\Pi_f(\psi[\vec{p}/\underline{\vec{p}}]) \quad &\text{iff} \quad \Upsilon\Pi_f(\psi) \to \Upsilon\Pi_f(\psi[\vec{p}/\underline{\vec{p}}]) \\ &\text{iff} \quad \forall\vec{p}\ \psi_f \to \psi_f[\vec{p}/\underline{\vec{p}}], \end{aligned}$$

where $\forall\vec{p}\ \psi_f$ is the optimised functional translation $\Upsilon\Pi_f(\psi)$ of ψ. $\forall\vec{p}\ \psi_f \to \psi_f[\vec{p}/\underline{\vec{p}}]$ is true, which proves the base case of (i).

(ii) The proof for the rules modus ponens and necessitation is as easy. $\square$

If there are other rules in Σ, we must prove case (ii) for these rules individually, which should not be a problem, in general.

3.3 Functional correspondence theory

For schemas, like 4, T, etcetera, that are definable by a first-order relational property the operator Υ has no effect. For example, the existential quantifier in the local associativity law

$$\forall x\ \forall\alpha\beta\exists\gamma\ [x\alpha\beta] = [x\gamma],$$

is already as far right as possible in the quantifier prefix. In some instances moving existential quantifiers inward increases the complexity of Skolem terms which is not desirable. Examples are the identities associated with the schema Mk:

$$\forall x\ \exists\alpha\ \exists\beta\ [x\alpha\beta] = x \land \forall x\ \forall\gamma\ \forall\delta\exists\epsilon\ [x\alpha\gamma\delta] = [x\epsilon].$$

For first-order definable complete logics, Corollary 3.2.1 is the important theorem of this chapter. Otherwise, as we will see, Theorem 3.3.2 is important and assists us in describing essentially second-order schemas by first-order functional formulae.

There are different methods for solving correspondence problems. The Sahlqvist technique is the most widely known method (van Benthem 1984). (Other methods developed more recently are by Szałas (1993), Simmons (1994).) We adopt the automated approach of Gabbay and Ohlbach (1992) who devised the SCAN algorithm, which attempts to reduce arbitrary existentially quantified second-order formulae to equivalent first-order formulae. This reduction may be applied to the second-order relational, functional or optimised functional translations of modal schemas.

Initially, we will consider how modal schemas can be reduced to their relational correspondence properties and why the reduction fails for McKinsey's schema. A formal description of SCAN can be found in Appendix A in the Appendix.

The derivation with SCAN of the correspondence property for any given modal schema φ requires as input the negation of its translation $\forall P_1 \dots P_n\psi$. Take for example the relational translation for the schema $T = \Box p \to p$.

$$\Pi_r(T) = \forall P\ \forall x(\forall y\ R(x,y) \to P(y)) \to P(x)).$$
$$\neg\Pi_r(T) = \exists P\ \exists x\ (\forall y\ R(x,y) \to P(y)) \land \neg P(x))$$

SCAN goes through three stages, namely forming the clausal form and Skolemising, exhaustively performing C-resolution on P-literals and when this stage terminates, the third stage attempts to reverse Skolemisation. The clausal form for our sample formula is:

1. $\neg R(\underline{x}, y), P(y)$
2. $\neg P(\underline{x})$.

The only C-resolution step upon P-literals yields

3. $\neg R(\underline{x}, \underline{x})$, $\qquad$ $[1, 2, C\text{-resolution}]$

the only clause that remains after purity deletion, which eliminates the original two clauses. Reversing the Skolemisation process we reintroduce quantifiers and get

$$\exists x\ \neg R(x,x).$$

This is negated again and the final result is

$$\forall x\ R(x,x).$$

The algorithm has reduced the second-order formula involving a P predicate and an R predicate associated with the schema T to its first-order equivalent formulation involving only the R predicate. A natural question is: Does SCAN reduce all first-order definable modal schemas to their characteristic properties of accessibility?

There are two critical points in the algorithm. One, saturation with respect to C-resolution may not terminate, and two, the reconstruction of the quantifiers for the Skolem functions may not be possible. For instance, the procedure produces infinitely many non-redundant clauses for the relational translation of the schema G. KG is determined by frames of finite R-chains, and this is not first-order definable.

McKinsey's schema is a simple schema not first-order definable in the relational context that illustrates a situation where reversing Skolemisation is impossible. (The critical schemas in the logic $\overline{K}_E$ with counting quantifiers that we will define in the next chapter have a very similar structure.) McKinsey's schema $M = \Box\Diamond p \to \Diamond\Box p$ translates to a universally quantified formula, which we negate to get

$$\begin{aligned} \exists P\,\exists x\; (\forall y_1\; R(x,y_1) \to (\exists y_2\; R(y_1,y_2) \wedge P(y_2))) \wedge \\ (\forall z_1\; R(x,z_1) \to (\exists z_2\; R(z_1,z_2) \wedge \neg P(z_2))). \end{aligned} \tag{3.4}$$

This is converted to the following set of clauses:

$$\begin{array}{ll} 1. & \neg R(\underline{x},y_1) \vee R(y_1,f(y_1)) \\ 2. & \neg R(\underline{x},y_1) \vee P(f(y_1)) \\ 3. & \neg R(\underline{x},z_1) \vee R(y_1,g(z_1)) \\ 4. & \neg R(\underline{x},z_1) \vee \neg P(g(z_1)). \end{array}$$

$\underline{x}$ is the Skolem constant for x, f is the Skolem function for y_2 and g is the Skolem function for z_2. As in the previous example, there is only one C-resolvent upon the predicate P (produced by resolving the clauses 2. and 4.). The final clause set after purity deletion is:

$$\begin{array}{ll} 1. & \neg R(\underline{x},y_1) \vee R(y_1,f(y_1)) \\ 3. & \neg R(\underline{x},z_1) \vee R(y_1,g(z_1)) \\ 5. & \neg R(\underline{x},y_1) \vee \neg R(\underline{x},z_1) \vee f(y_1) \neq g(z_1). \end{array}$$

It is not possible to reconstruct the existential quantifiers for the Skolem functions f and g producing a linear ordering of quantifiers. The problem is that f depends on y_1 and g depends on z_1 and both $f(y_1)$ and $g(z_1)$ occur in the same clause (namely 5.). There is a way of reconstructing quantifiers for f and g by means of parallel Henkin quantifiers:

$$\exists x \begin{pmatrix} \forall y_1 \exists y_2 \\ \forall z_1 \exists z_2 \end{pmatrix} \begin{array}{r} (R(x,y_1) \to R(y_1,y_2)) \wedge (R(x,z_1) \to R(z_1,z_2)) \\ \wedge\, ((R(x,y_1) \wedge R(x,z_1)) \to y_2 \neq z_2). \end{array}$$

Unfortunately, Henkin quantifiers are not first-order quantifiers. Since the formula must be negated in order to get the frame property for the McKinsey schema and it is not clear how negated Henkin quantifiers are defined, this form is useless for the purposes of automated reasoning.

If we were allowed to change the variable dependency for f and g in a suitable way *reversing Skolemisation* is possible. For (3.4) suitably changing the variable dependency means, moving the existential quantifiers outward over the universal quantifiers:

$$\begin{aligned} \exists P\,\exists x\; (\exists y_2 \forall y_1\; R(x,y_1) \to (R(y_1,y_2) \wedge P(y_2))) \wedge \\ (\exists z_2 \forall z_1\; R(x,z_1) \to (R(z_1,z_2) \wedge \neg P(z_2))). \end{aligned}$$

If we apply C-resolution to this formula we get

$$\begin{array}{ll} 1. & \neg R(\underline{x}, y_1) \vee R(y_1, f) \\ 3. & \neg R(\underline{x}, z_1) \vee R(y_1, g) \\ 5. & \neg R(\underline{x}, y_1) \vee \neg R(\underline{x}, z_1) \vee f \neq g \end{array}$$

and Skolemisation can be reversed for this set bearing a linearly quantified (first-order) formula which we can negate without problems. But, of course, moving existential quantifiers outward, and in particular, over universal quantifiers, is in general not admissible. The operation does not preserve logical equivalence.

In maximal functional frames the situation is such that moving existential quantifiers outward in the negation of translated schemas is admissible. Above in (3.3), we derived $\neg\Upsilon\Pi_f(M)$ for McKinsey's schema:

$$(*) \qquad \exists P\ \exists x\ \exists\beta\beta'\forall\alpha\alpha'\ P[x\alpha\beta] \wedge \neg P[x\alpha'\beta']$$

and its clause form:

$$\begin{array}{l} 1.\ P[\alpha\underline{\beta}] \\ 2.\ \neg P[\alpha'\underline{\beta}']. \end{array}$$

C-resolution yields:

$$[\alpha\underline{\beta}] \neq [\alpha'\underline{\beta}'].$$

Reversing Skolemisation yields:

$$\exists x\ \exists\beta\beta'\forall\alpha\alpha'\ [x\alpha\beta] \neq [x\alpha'\beta']$$

and negation produces:

$$\forall x\ \forall\beta\beta'\ \exists\alpha\alpha'\ [x\alpha\beta] = [x\alpha'\beta']. \tag{3.5}$$

This is first-order and equivalent to $(*)$. The corresponding theory equation is that of Figure 3.4.

The operation Υ moves all existential quantifiers inward as far as possible, which is more than the theory demands. For dealing with McKinsey's schema swapping one quantifier suffices. The resulting formula is

$$\forall x\ \forall\beta\exists\alpha\forall\beta'\exists\alpha'\ [x\alpha\beta] = [x\alpha'\beta']$$

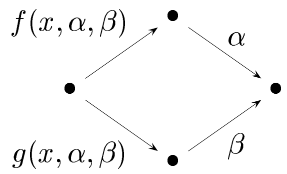
and the associated theory equation is

$$[xf(x,\beta)\beta] = [xg(x,\beta,\beta')\beta'].$$

This is slightly stronger than (3.5), but it is still first-order.

Although KM is not complete with respect to a class of frames described by a first-order property of a relation, KM is complete with respect to some class of frames, which Fine (1975) defined via normal forms. Therefore, by Theorem 3.2.2, Theorem 2.3.1 and the fact that $\Upsilon\Pi_f(M)$ is a first-order property:

$M \quad [xf(x,\alpha,\beta)\alpha] = [xg(x,\alpha,\beta)\beta]$



$F \quad [xf(x,\alpha,\beta)] = [xg(x,\alpha,\beta)\alpha] \wedge [xf(x,\alpha,\beta)] = [xg'(x,\alpha,\beta)\beta]$

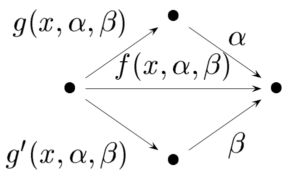


Figure 3.4: 'Optimised' theory equations.

Corollary 3.3.1 A modal formula φ is a theorem of *KM* iff

$$(\forall x\ \forall\beta\beta'\ \exists\alpha\alpha'\ [x\alpha\beta] = [x\alpha'\beta']) \rightarrow \Upsilon\Pi_f(\varphi) \quad \text{is a first-order theorem.}$$

Since in the clausal forms no Skolem functions appear except for Skolem constants, reversing Skolemisation is always possible, when the C-resolution procedure terminates. Theorem 3.2.2 has a corollary useful for practical purposes, namely:

Theorem 3.3.2 Let *K(D)*Σ be a complete (respectively complete and first-order definable) modal logic with modus ponens and the necessitation rule being the only rules. For any modal formula φ, if SCAN is successful in eliminating all second-order quantifiers for all schemas from $\neg\Upsilon\Pi_f(\Sigma)$ then

$$\varphi \text{ is a theorem in } K(D)\Sigma \quad \text{iff} \quad \Upsilon\Pi_f(\Sigma) \rightarrow \Upsilon\Pi_f(\varphi) \text{ is a first-order theorem}$$

The result transfers also to multi-modal logics $K_{(m)}\Sigma$. The result is remarkable, for no similar result holds for the relational translation.

A sample derivation demonstrates deduction by resolution and paramodulation for McKinsey's schema. The formula

$$F \quad (\Box\Diamond p \wedge \Box\Diamond q) \rightarrow \Diamond(p \wedge q)$$

is provable in *K4M* (a conventional proof can be found in Hughes and Cresswell 1996, pp. 131). Since *KM* is serial the clausal form of $\neg\Upsilon\Pi_f(F)$ is:

1. $P[\alpha\underline{\beta}]$
2. $Q[\alpha\underline{\gamma}]$
3. $\neg P[\alpha] \vee \neg Q[\alpha]$

The paramodulant of 3. with local associativity is

4. $\neg P[\alpha\beta] \vee \neg Q[\alpha\beta]$.

With $[xf(x, \alpha, \beta)\alpha] = [xg(x, \alpha, \beta)\beta]$ a paramodulant is

$$5.\ \neg P[g([], \alpha, \beta)\alpha] \vee \neg Q[g([], \alpha, \beta)\alpha].$$

Now, resolve with 1. to get

$$6.\ \neg Q[g([], \alpha, \underline{\beta})\alpha].$$

The empty clause follows then from 6. and 2. We can also infer $\Upsilon\Pi_f(M)$ from $\Upsilon\Pi_f(F)$: Since KF is serial, the theory approximating it is the conjunction of two identities from Figure 3.4. The clausal form for M is:

$$1.\ P[\alpha\underline{\beta}]$$
$$2.\ \neg P[\alpha\underline{\gamma}]$$

The proof is:

$$3.\ P[f(\underline{\beta}, \alpha)]$$
$$4.\ P[g'([], \underline{\beta}, \alpha)\alpha]$$
$$5.\ \emptyset$$

by paramodulating into 1. and 3., and resolving 2. and 4. Cross checking, we note M is logically equivalent to the instance of F with $q = \neg p$.

3.4 Conclusion

The functional language is in a sense more expressive than the relational language and therefore properties of frames which are second-order in the relational language may become first-order in the functional language. It must be stressed that we only showed that we can prove that a formula φ is a $K\Sigma$ theorem by proving a weaker theorem from weaker (possibly first-order) frame properties, whereas the original (equivalent) frame properties are still second-order, even in the functional language. But from a practical and theorem proving point of view, we achieved the desired effect, in that we approximated second-order frame properties by first-order frame properties without changing the logic.

In summary, thus far the functional translation of modal formulae has been investigated for the theorems to be proved in some first-order definable modal logics. In the case of propositional modal logic, it is possible to move in the functional translation of the negated non-schemas all existential quantifiers over universally quantified 'accessibility functions' to the front. The effect is that complex Skolem terms are avoided and at most Skolem constants occur in the clausal form. This is not new. New is, that the same can be said for the functional translation of modal schemas. We demonstrated that McKinsey's schema which is not first-order in the relational context has a corresponding first-order functional approximation. This formulation can be computed automatically using a quantifier elimination algorithm like SCAN, if quantifiers are suitably swapped. This is not possible in the relational language. We have shown that swapping the existential and the universal quantifiers in the functional language preserves the theorems of the logic, provided the quantifiers are also swapped in the theorem which we wish to prove.

Using the methods proposed in this chapter, we can now apply first-order predicate logic theorem proving techniques to a wider class of modal systems than was possible before.

Recent incompleteness results found by Gasquet (1994) suggest that optimisation cannot be readily applied to quantified modal logics. In quantified modal logics the truth of a formula does depend on predecessor worlds and the equivalence (3.1) does not hold. An example (due to A. Herzig) shows what can happen. The formula $\Box(\exists z\ (p(z) \wedge \Box\Diamond\neg p(z)))$ is true at the world x of the following model.

$$\begin{tikzcd} & y_1 \quad p(a), \neg p(b) \\ x \arrow[ur, leftrightarrow] \arrow[dr, leftrightarrow] & \\ & y_2 \quad p(b), \neg p(a) \end{tikzcd}$$

For every world y accessible from x (these are y_1 and y_2) there is a world in which $p(z)$ holds (for y_1, z is a and for y_2, z is b), and for every worlds y' accessible from y (x) there is a world y'' accessible from y' (y_1 and y_2 are the candidates) such that $\neg p(z)$ holds at y''. Now we have to choose either y_1 or y_2 and check whether $\neg p(z)$ holds, but z was determined in a previous world, in case that $y = y_1$, z is a and in case that $y = y_2$, z is b. Our choice depends on the path we choose to get to $y' = x$. Therefore, if there are domain quantifiers, moving existential quantifiers to the front is not always possible. For quantified modal logic the technique does not work in general (neither does it for relational frames or functional frames). It remains to be investigated whether exchanging quantifiers is possible at least for some cases, for example, for the case that domain variables do not occur in different modal contexts.

Chapter 4

Translating graded modal logic

In the logic of graded modalities it is possible to talk about sets of finite cardinality. Various calculi exist for graded modal logics and all generate vast amounts of case distinctions. In this chapter we transform graded modal logic into clausal form by the optimised translation method via a second intermediary multi-modal logic. The translation is sound and complete. In contrast to known approaches the resulting resolution calculus enables us to reason with cardinalities of sets symbolically. We will see in many cases the lengths of our proofs are independent of the cardinalities. We consider the basic uni-modal graded modal logic $\overline{K}$ which we briefly review in Section 4.1. This logic will be accommodated in a normal multi-modal logic, called $\overline{K}_E$, that is introduced in Section 4.2 and the embedding of $\overline{K}$ in $\overline{K}_E$ is described in Section 4.3. The initial reduction to first-order logic by the optimised functional translation is then straightforward and yields a set of theory clauses that define the original logic $\overline{K}$. However, things are not all that simple as the sample derivations in Section 4.4 will show. The material of this chapter is published in Ohlbach et al. (1996).

4.1 The graded modal logic $\overline{K}$

The box operator and the diamond operator of normal modal logics are also called the necessity and possibility operators. In (1970) Goble defines modal logics with a fixed and finite number of box modalities that each represent a different grade of necessity. For example, the formula $N_m\varphi \wedge N_n\psi$ for positive integers $m < n$, is read to mean ψ is more necessary than φ. Fine (1969, 1972) generalises this idea and introduces modal logics with *numerical modalities*. These are now commonly referred to as modal logics with *graded modalities*. In a series of papers Fattorosi-Barnaba, de Caro and Cerrato (1985, 1988, 1988, 1990) rediscover and analyse these logics. Recent investigations of graded modal logics are by van der Hoek (1992). Together with de Rijke he applies graded modal logic to linguistics and artificial intelligence. In (1993b) they show that generalised quantifiers can be modelled with graded modalities. In (1993a, 1995) they also show that certain numerical quantifier operations available in KL-ONE-based knowledge representation languages can be modelled with graded modalities.

We adopt the definition of the graded modal logic $\overline{K}$ of van der Hoek (1992). $\overline{K}$ extends the basic modal logic K with graded modalities. Formally, the vocabulary of $\overline{K}$ is that of propositional logic consisting of finitely many propositional symbols $p, q, \ldots, \perp$ and $\rightarrow$, plus countably many modal operator symbols $\blacklozenge_n$ for $n \in \mathbb{N}_0$. Formulae of $\overline{K}$ have the following forms:

$$p, q, \dots, \quad \bot, \quad \varphi \to \psi, \quad \blacklozenge_n \varphi.$$

Negation, disjunction, conjunction, etcetera, are defined as usual in terms of bottom and implication. Additional modal operators are $\blacksquare_n$, $\blacklozenge!_0$ and $\blacklozenge!_n$, defined by

$$\begin{aligned}
&\blacksquare_n \varphi = \neg \blacklozenge_n \neg \varphi \quad \text{for } n \geq 0,\\
&\blacklozenge!_0 \varphi = \blacksquare_0 \neg \varphi \quad \text{and}\\
&\blacklozenge!_n \varphi = \blacklozenge_{n-1} \varphi \wedge \neg \blacklozenge_n \varphi \quad \text{with } n > 0.
\end{aligned}$$

$\blacklozenge_n \varphi$ is read to mean 'φ is true in *more than* n accessible worlds', $\blacksquare_n \varphi$ is read to mean '$\neg\varphi$ is true in *at most* n accessible worlds', and $\blacklozenge!_n \varphi$ is read to mean 'φ is true in *exactly* n accessible worlds'.

The logic $\overline{K}$ of graded modalities is defined by the following schemas

$A1$	the schemas of propositional logic
$A2$	$\blacklozenge_{n+1} p \to \blacklozenge_n p$
$A3$	$\blacksquare_0 (p \to q) \to (\blacklozenge_n p \to \blacklozenge_n q)$
$A4$	$\blacksquare_0 \neg (p \wedge q) \to ((\blacklozenge!_n p \wedge \blacklozenge!_m q) \to \blacklozenge!_{n+m} (p \vee q))$

together with the rules of modus ponens and necessitation for $\blacksquare_0$.

Let ∇ denote exclusive-or. The following are theorems of $\overline{K}$ (van der Hoek 1992).

$A5$	$\blacksquare_0 (p \to q) \to (\blacksquare_n p \to \blacksquare_n q)$
$A6$	$\blacklozenge_n (p \wedge q) \to (\blacklozenge_n p \wedge \blacklozenge_n q)$
$A7$	$(\blacklozenge!_n p \wedge \blacklozenge!_m p) \to \bot \qquad$ for $n \neq m$
$A8$	$\blacksquare_n \neg p \leftrightarrow (\blacklozenge!_0 p \ \nabla\ \blacklozenge!_1 p \ \nabla\ \dots \ \nabla\ \blacklozenge!_n p)$
$A9$	$\neg \blacklozenge_n (p \vee q) \to \neg \blacklozenge_n p$
$A10$	$\blacklozenge_{n+m} (p \vee q) \to (\blacklozenge_n p \vee \blacklozenge_m q)$
$A11$	$(\blacklozenge!_n p \wedge \blacklozenge_m p) \to \bot \qquad$ for $m \geq n$
$A12$	$\blacklozenge_n (p \wedge q) \wedge \blacklozenge_m (p \wedge \neg q) \to \blacklozenge_{n+m+1} p$

Observe that $\blacklozenge_0$ and $\blacksquare_0$ coincide with the standard modal operators $\Diamond$ and $\Box$. K is therefore a subsystem of $\overline{K}$.

The semantics of $\overline{K}$, like K, is given by a frame (W, R) consisting of a non-empty set W of worlds and a binary accessibility relation R over W. Any frame gives rise to a class of models $\mathcal{M} = (W, R, \iota)$ by adjoining a valuation function ι mapping propositional variables to subsets of W. *Truth* in a model $\mathcal{M}$ for formulae of $\overline{K}$ at any world x is defined as for K except the truth of modal formulae is specified by:

$$\begin{aligned}
&\mathcal{M}, x \models \blacklozenge_n \varphi \quad \text{iff} \quad \mathrm{card}(\{y \in W \mid R(x,y) \wedge \mathcal{M}, y \models \varphi\}) > n\\
&\mathcal{M}, x \models \blacksquare_n \varphi \quad \text{iff} \quad \mathrm{card}(\{y \in W \mid R(x,y) \wedge \mathcal{M}, y \models \neg\varphi\}) \leq n.
\end{aligned} \tag{4.1}$$

Recall, $R''(x) = \{y \mid R(x,y)\}$. Then, satisfiability of $\blacklozenge_n \varphi$ and $\blacksquare_n \varphi$ can be reformulated as follows:

$$\begin{aligned}
&\mathcal{M}, x \models \blacklozenge_n \varphi \quad \text{iff} \quad \exists Y \subseteq R''(x) \text{ with } \mathrm{card}(Y) > n \text{ and } \forall y \in Y : \mathcal{M}, y \models \varphi\\
&\mathcal{M}, x \models \blacksquare_n \varphi \quad \text{iff} \quad \forall Y \subseteq R''(x) \text{ with } \mathrm{card}(Y) > n,\ \exists y \in Y : \mathcal{M}, y \models \varphi.
\end{aligned} \tag{4.2}$$

(The proof is routine.) It is now easy to verify $\blacklozenge_n\varphi \leftrightarrow \neg\blacksquare_n\neg\varphi$.

The axiomatisation of $\overline{K}$ is sound and complete: For any formula φ,

$$\varphi \text{ is provable in } \overline{K} \quad \text{iff} \quad \varphi \text{ is valid in all frames.}$$

A proof can be found in de Caro (1988) (which improves the proof of Fattorosi-Barnaba and de Caro 1985).

In the remainder of this chapter we assume the formulae of $\overline{K}$ to be in *negation normal form* which can be obtained by systematically applying the following equivalences from left to right.

$$\begin{array}{rl} \neg(\varphi \vee \psi) \leftrightarrow (\neg\varphi \wedge \neg\psi) & \qquad \varphi \leftrightarrow \psi \leftrightarrow ((\varphi \rightarrow \psi) \wedge (\psi \rightarrow \psi)) \\ \neg(\varphi \wedge \psi) \leftrightarrow (\neg\varphi \vee \neg\psi) & \qquad \neg\blacksquare_n\varphi \leftrightarrow \blacklozenge_n\neg\varphi \\ \varphi \rightarrow \psi \leftrightarrow (\neg\varphi \vee \psi) & \qquad \neg\blacklozenge_n\varphi \leftrightarrow \blacksquare_n\neg\varphi \end{array}$$

4.2 The multi-modal logic $\overline{\boldsymbol{K}}_E$

In this section we define the more expressive multi-modal logic with numeric quantification, called $\overline{K}_E$, that encompasses $\overline{K}$. $\overline{K}_E$ is used as an intermediary logic for transforming the graded modal logic into first-order logic with the aim of facilitating inference for $\overline{K}$ by standard theorem proving methods based on the functional translation. In contrast to $\overline{K}$, $\overline{K}_E$ is a normal modal logic.

$\overline{K}_E$ has two kinds of modalities $\Diamond_n$ and $\Diamond$ and is an attempt at capturing frames of the form

$$(W \cup W^Y, \{R_n\}_{n\in\mathbb{N}_0}, E) \tag{4.3}$$

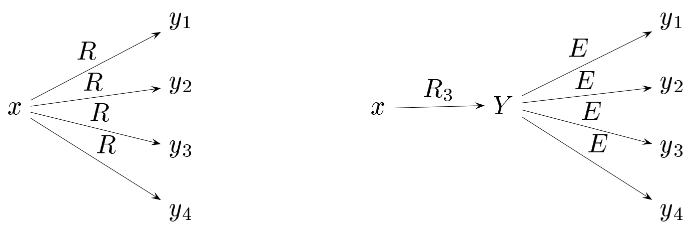
with W^Y a subset of the power set $\mathbf{2}^W$ and two kinds of accessibility relations R_n and E. R_n relates elements of W with subsets Y of W that have cardinality greater than n, and E relates subsets Y of W to elements of Y. E can be thought as being the converse of the 'element of' relationship. The sets Y are in W^Y.

Frames of $\overline{K}$ of the form (W, R) can then be embedded in frames of the form (4.3) by replacing R by the family $\{R_n\}_{n\in\mathbb{N}_0}$ together with E, and W by its superset $W \cup W^Y$ that includes also subsets of W. For example, according to the definition of the previous section, $\blacklozenge_3\varphi$ is true in a world x iff there are at least 4 worlds to which x is related by R. This definition is depicted in the Figure .1(a). Figure .1(b) depicts its encoding in a frame of $\overline{K}_E$. The relation R is encoded by the relational composition of the two new relations R_3 and E. There is a new world labelled Y which is meant to represent the set of worlds y_1, y_2, y_3 and y_4.

$\overline{K}_E$ differs from an alternative translation into a multi-modal logic called 'Lcount', developed by Andréka, Németi and Sain (1995). In their system there are n-place operators $\Diamond_n$ with semantics

$$\begin{array}{lll} \mathcal{M}, x \models \Diamond_n\varphi_1, \ldots, \varphi_n & \text{iff} & \text{there are distinct } y_1, \ldots, y_n \text{ such that} \\ & & \mathcal{M}, y_1 \models \varphi_1 \text{ and } \ldots \text{ and } \mathcal{M}, y_n \models \varphi_n. \end{array}$$

Calculi based on this semantics, however, seem not to be much different from the calculi based on the original semantics for graded modalities. In a corresponding tableaux system one has to introduce witnesses for the worlds again, but this is what we want to avoid.



(a) (W, R) (b) $(W \cup W^Y, \{R_n\}_{n\in\mathbb{N}_0}, E)$

Figure 4.1: Encoding a $\overline{K}$ frame as a $\overline{K}_E$ frame

Formally, the language of $\overline{K}_E$ is that of $\overline{K}$ with the operator $\blacklozenge_n$ replaced by the operators $\Diamond_n$ and $\Diamond$. Formulae of $\overline{K}_E$ have the following forms:

$$p, q, \dots, \quad \bot, \quad \varphi \to \psi, \quad \Diamond_n\varphi, \quad \Diamond\varphi.$$

The classical connectives are defined in the usual way. The duals of $\Diamond_n$ and $\Diamond$ are $\Box_n = \neg\Diamond_n\neg$ and $\Box = \neg\Diamond\neg$. The intended meaning of $\Diamond_n\varphi$ (respectively $\Diamond\varphi$) is the standard one, namely φ is true in some world accessible by the binary relation R_n (resp. E). We call $\Diamond_n$ and $\Box_n$ the *numerical operators* and $\Diamond$ and $\Box$ the *membership operators*. The relations R_n and E are defined as sketched above. Namely, W forms the domain of the R_n and the co-domain of E and the new class of worlds W^Y forms the co-domain of the R_n and the domain of E.

$\overline{K}$-formulae of the form $\blacklozenge_n\varphi$ and $\blacksquare_n\varphi$ can be formulated as $\overline{K}_E$-expressions of the form

$$\Diamond_n\Box\varphi \quad \text{and} \quad \Box_n\Diamond\varphi,$$

respectively (this can be seen easily using 4.2). The logic $\overline{K}_E$ is more expressive than $\overline{K}$. Nevertheless, it has similar properties as $\overline{K}$. $\overline{K}_E$ permits arbitrary combinations of modal operators, not only alternate combinations of necessity and possibility operators. For example, $\Box_3\Box_4\Box\Box\varphi$ is a well-formed formula of $\overline{K}_E$, although it may not make much sense in our intended semantics. However, there are combinations of modal operators which have no equivalent formulation in $\overline{K}$, but which have interesting applications. Here is an example of such a formula:

$$\Box_{10}(\textit{soccer-team} \to \Box\textit{soccer-player}).$$

It says that, every set with more than 10 elements that is a soccer team contains only soccer players. In this way we can distinguish between notions like teams which we interpret as sets and notions like players which we interpret as elements.[1]

We now give an axiomatisation for $\overline{K}_E$ and investigate its characteristic frames. The axiom schemas and rules are: For any $n, m \in \mathbb{N}_0$

$N1$ the axiom schemas of propositional logic and modus ponens

$N2$ the K-schemas for $\Box_n$ and $\Box$:
$\Box_n(p \to q) \to (\Box_n p \to \Box_n q) \qquad \Box(p \to q) \to (\Box p \to \Box q)$

[1]In linguistic applications this distinction may be put to use as proposed in Schmidt (1997b).

$N3$ the necessitation rules for $\Box_n$ and $\Box$:
If $\vdash p$ then $\vdash \Box_n p$ if $\vdash p$ then $\vdash \Box p$

$N4$ $\Box_0 \Diamond p \to \Box_n \Box p$

$N5$ $\Diamond_n \Box p \to \Diamond_n \Diamond p$

$N6$ $\Box_n p \to \Box_{n+1} p$

$N7$ $\Diamond_{n+m} \Box (p \vee q) \to (\Diamond_n \Box p \vee \Diamond_m \Box q)$

$N8$ $(\Diamond_n \Box (p \wedge q) \wedge \Diamond_m \Box (p \wedge \neg q)) \to \Diamond_{n+m+1} \Box p$

$N1$–$N3$ are basic in every normal modal logic. $N4$ captures that, if something holds for every set of worlds with more than zero elements, that is, if it holds for every non-empty set of worlds, then it holds also for all sets with more than n elements. This means, the composition $R_n\,;E$ (for n arbitrary) is a subrelation of the composition $R_0\,;E$. $N5$ ensures that no set with more than n elements is empty. A contrapositive version of $N6$ is $\Diamond_{n+1} p \to \Diamond_n p$. It captures that sets with more than $n+1$ elements are sets with more than n elements.

$N7$ corresponds to $A10$ and is a bit more complicated to explain. As an example suppose $n = 2$ and $m = 4$. For these values $N7$ is $\Diamond_6 \Box (p \vee q) \to (\Diamond_2 \Box p \vee \Diamond_4 \Box q)$ which is equivalent to

$$(\Diamond_6 \Box (p \vee q) \wedge \neg \Diamond_2 \Box p) \to \Diamond_4 \Box q.$$

In words, if there are more than 6, say 7, worlds in which the formula $p \vee q$ is true, but it is not the case that p holds in more than two worlds (that is, $\neg p$ is true in all but possibly two worlds) then in the remaining 5 of 7 worlds q is true. The schema says that every $(n+m)$-element set can be decomposed into an n-element set and an m-element set, but note, the schema is slightly stronger.

The intuition underlying $N8$ is the following. Suppose there is a set Y_1 with at least $n+1$ elements where $p \wedge q$ holds and there is another set Y_2 with at least $m+1$ elements where $p \wedge \neg q$ holds. Since q and $\neg q$ cannot hold simultaneously in one world, Y_1 and Y_2 must be disjoint. Thus, p holds in $Y_1 \cup Y_2$ which is of cardinality at least $n+m+2$. Therefore, $\Diamond_{n+m+1} \Box p$ is true.

Now we turn to the semantics of $\overline{K}_E$. The K-schemas and the necessitation rules allow us to use the standard Kripke semantics. A $\overline{K}_E$ *frame* is a relational structure

$$\mathcal{F} = (W, \{R_n\}_{n \in \mathbb{N}_0}, E).$$

W is a non-empty set of worlds. The R_n are binary relations over W and E is a designated binary relation over W. R_n and E satisfy the properties $N4'$–$N8'$ given below. A *model* of $\overline{K}_E$ *based on a frame* $\mathcal{F}$ is a pair $\mathcal{M} = (\mathcal{F}, \iota)$ with ι defined as before. Truth is defined as in any multi-modal logic with R_n determining truth for $\Diamond_n$ and E determining truth for $\Diamond$.

The following are the characteristic properties of $\overline{K}_E$-frames that correspond to the schemas $N4$–$N8$: For any $n, m \in \mathbb{N}_0$

$N4'$ $\forall xyz\ ((R_n(x,y) \wedge E(y,z)) \to \exists u\ (R_0(x,u) \wedge \forall v\ (E(u,v) \to v = z)))$

$N5'$ $\forall xy\ (R_n(x,y) \to \exists z\ E(y,z))$

$N6'$ $\forall xy\ (R_{n+1}(x,y) \to R_n(x,y))$

$N7'$ $\forall xy\ R_{n+m}(x,y) \to \forall fg\ \exists uv (R_n(x,u) \to E(u,f(u)) \wedge R_m(x,v) \to$
$E(v,g(v))) \to (R_n(x,u) \wedge R_m(x,v) \wedge E(y,f(u)) \wedge f(u) = g(v))$

$$\begin{array}{ll} N8' & \forall xyz\ (R_n(x,y) \wedge R_m(x,z) \wedge \forall u\ (E(y,u) \rightarrow \neg E(z,u)) \rightarrow \\ & \quad \exists v\ (R_{n+m+1}(x,v) \wedge \forall w\ (E(v,w) \rightarrow E(y,w) \vee E(z,w)))). \end{array}$$

We computed these properties with SCAN, which guarantees soundness of the semantics with respect to the axiomatisation, that is,

$$\text{if } \vdash_{\overline{K}_E} \varphi \text{ then } \models_{\overline{K}_E} \varphi. \tag{4.4}$$

If each of the properties $N4'$–$N8'$ is first-order then the Sahlqvist Theorem (Sahlqvist 1975, van Benthem 1983, van Benthem 1984) ensures completeness for $\overline{K}_E$. Unfortunately, $N7'$ is still second-order, which forces us to prove completeness explicitly. We do this indirectly *for translated $\overline{K}$ formulae* by using the completeness of $\overline{K}$ and the soundness and completeness of the translation into $\overline{K}_E$ which will be proved in the next section. General completeness for arbitrary formulae of $\overline{K}_E$ is open,[2] but for the purpose of our translation, this is not detrimental.

The correspondence property $N4'$ states that all singleton subsets of the set of worlds accessible by R_n are uniquely represented by a world accessible by R_0. $N5'$ asserts that every world accessible by R_n leads via E to another world. We say E is *weakly serial.* By $N6'$ the set $\{R_n\}_{n\in\mathbb{N}_0}$ of R_n relations forms a linear order with R_0 being the largest element, since for any $m > n$, R_m is a subrelation of R_n.

The correspondence property $N7'$ of $N7$ expresses intuitively that every set y with more than $n+m$ elements can be decomposed into a set u with more than n elements and a set v with more than m elements, and if y happens to have exactly $n+m+1$ elements then u and v overlap in at least one element.

$N8'$ expresses, as already mentioned, that for disjoint sets the cardinality of their union is the sum of the cardinalities of the sets.

For a better understanding of the frame properties it is helpful to think of the variable y in $R_n(x,y)$ and $E(y,z)$ as representing a set Y, $R_n(x,y)$ as representing that the cardinality of Y is greater than n, and $E(y,z)$ as representing that z is an element of Y. Then $N4'$–$N8'$ represent:

$$\begin{array}{ll} N4'' & \forall Y \forall z\ (((\mathrm{card}(Y) > n \wedge z \in Y) \rightarrow \{z\} \subseteq Y) \\ N5'' & \forall y\ (\mathrm{card}(Y) > n \rightarrow Y \neq \emptyset) \\ N6'' & \forall Y\ (\mathrm{card}(Y) > n+1 \rightarrow \mathrm{card}(Y) > n) \\ N7'' & \forall Y\ (\mathrm{card}(Y) > n+m \rightarrow \forall fg\ \exists UV\ ((\mathrm{card}(U) > m \wedge \mathrm{card}(V) > m) \rightarrow \\ & \quad \text{if } f \text{ selects from } U \text{ and } g \text{ from } V \text{ then } f(U) \in Y \wedge f(U) = g(V))) \\ N8'' & \forall YZ\ ((\mathrm{card}(Y) > n \wedge \mathrm{card}(Z) > m \wedge Y \cap Z = \emptyset) \rightarrow \\ & \quad \exists V\ (\mathrm{card}(V) > n+m+1 \wedge V \subseteq Y \cup Z)). \end{array}$$

We can show that the frames in the standard class associated with $\overline{K}_E$ have the expected structure, namely that all worlds accessible by R_n have more than n successors by E. However, non-standard $\overline{K}_E$-frames exist which do not have this intended structure. The problem is, we cannot enforce in a normal modal axiomatisation that R_1-accessible worlds have *more than* one E-successor. This may be captured by a schema like

$$\Box_1(\exists p\ (\Diamond p \wedge \Diamond \neg p)),$$

[2] See (i) in Section 4.5

or a rule similar to Gabbay's (1981) irreflexivity rule (but this gives no new theory). See also Prior (1968). The modal language of $\overline{K}$ and $\overline{K}_E$ is not expressive enough to characterise this class of frames. On the other hand, we can show using an inductive argument that whenever an R_1-successor has more than one E-successor, then for any positive integer n every R_n-successor has more than n E-successors. This is to say, the induction step goes through, but unfortunately the base case of the induction cannot be guaranteed. Because the translation of the logic $\overline{K}$ into the logic $\overline{K}_E$ is sound and complete (which we show in the next section), we know whenever a translated $\overline{K}$-formula has a model then it has a model with the expected structure.

We do not investigate the non-standard models further. It may turn out that they are p-morphic images of standard models, in which case they are completely irrelevant because normal modal logics cannot distinguish p-morphic images.

4.3 Translating $\overline{K}$ to $\overline{K}_E$

Now we formally define a translation function Π mapping formulae of $\overline{K}$ to formulae of $\overline{K}_E$ and show the translation is sound and complete.

Π maps $\overline{K}$-formulae to $\overline{K}_E$-formulae according to the following constraints. For any propositional variable p and any formulae φ and ψ,

$$\Pi(p) = p \qquad \Pi(\blacklozenge_n \varphi) = \Diamond_n \Box\, \Pi(\varphi)$$
$$\Pi(\neg\varphi) = \neg\Pi(\varphi) \qquad \Pi(\blacksquare_n \varphi) = \Box_n \Diamond\, \Pi(\varphi),$$
$$\Pi(\varphi \mathbin{@} \psi) = \Pi(\varphi) \mathbin{@} \Pi(\psi)$$

where @ denotes any binary classical connective $\wedge$, $\vee$, $\to$ or $\leftrightarrow$.

Theorem 4.3.1 (Soundness of Π) The translation Π from $\overline{K}$ into $\overline{K}_E$ is sound. That is, for any formula φ of $\overline{K}$

$$\text{if } \varphi \text{ is a theorem in } \overline{K} \text{ then } \Pi(\varphi) \text{ is a theorem in } \overline{K}_E.$$

Proof. Suppose φ is a theorem in $\overline{K}$. We proceed by induction on the length of the proof of φ and show that the proof sequence of φ in $\overline{K}$ determines a proof sequence of $\Pi(\varphi)$ in $\overline{K}_E$. We are done if we can show that the Π-translations of the schemas and the rules of $\overline{K}$ are $\overline{K}_E$-theorems.

Π leaves the propositional schemas and modus ponens unchanged. The translation of the necessitation rule N is:

$$\vdash_{\overline{K}_E} \varphi \quad \text{implies} \quad \vdash_{\overline{K}_E} \Box_0\Diamond\varphi.$$

If φ holds then, by the necessitation rule for $\Box$ and $\Box_0$, $\Box_0\Box\varphi$ holds. Apply modus ponens using the contrapositive instance with $n=0$ of $N5$ and get $\Box_0\Diamond\varphi$. (Formally, instead of φ one has to consider $\Pi(\varphi)$. But for the proofs this makes no difference.)

The translation of $A2$ is a contrapositive version of $N6$. It remains to prove the translations of $A3$ and $A4$ can be derived from the axiom schemas of $\overline{K}_E$ by using its rules.

For $A3$ we prove

$$\Pi(A3) = \Box_0\Diamond(p \to q) \to (\Diamond_n\Box p \to \Diamond_n\Box q)$$

is a theorem in $\overline{K}_E$. Suppose $\Box_0\Diamond(p \to q)$ and $\Diamond_n\Box p$ hold. Suppose further that $\neg\Diamond_n\Box q$, that is, $\Box_n\Diamond\neg q$, holds. From $\Box_0\Diamond(p \to q)$ we infer by $N4$ that $\Box_n\Box(p \to q)$ holds. By schema K for $\Box$, it follows that $\Box_n(\Box p \to \Box q)$ holds. This is equivalent to $\Box_n(\neg\Box q \to \neg\Box p)$, that is, $\Box_n(\Diamond\neg q \to \Diamond\neg p)$. Using K for $\Box_n$ we infer $\Box_n\Diamond\neg q \to \Box_n\Diamond\neg p$. From $\neg\Diamond_n\Box q$, which is equivalent to $\Box_n\Diamond\neg q$, using modus ponens we get $\Box_n\Diamond\neg p$, or equivalently $\neg\Diamond_n\Box p$. This contradicts $\Diamond_n\Box p$. Thus $\Pi(A3)$ is derivable in $\overline{K}_E$.

For $A4$: Let

$$\phi = \Box_0\Diamond\neg(p \wedge q) \wedge \Diamond_{n-1}\Box p \wedge \neg\Diamond_n\Box p \wedge \Diamond_{m-1}\Box q \wedge \neg\Diamond_m\Box q.$$

Then $\Pi(A4)$ is equivalent to

$$\phi \to (\Diamond_{n+m-1}\Box(p \vee q) \wedge \neg\Diamond_{n+m}\Box(p \vee q)).$$

We prove this in two steps. First, we prove $\phi \to \Diamond_{n+m-1}\Box(p \vee q)$. Suppose ϕ holds. It suffices to show

$$\Diamond_{n-1}\Box(p \wedge \neg q). \tag{4.5}$$

From $\Diamond_{n-1}\Box(p \wedge \neg q)$, or equivalently $\Diamond_{n-1}\Box((p \vee q) \wedge \neg q)$, and $\Diamond_{m-1}\Box q$, or equivalently $\Diamond_{m-1}\Box((p \vee q) \wedge q)$, using $N8$ we deduce $\Diamond_{n+m-1}\Box(p \vee q)$.

For proving that (4.5) follows from ϕ we proceed by contradiction. Suppose that $\neg\Diamond_{n-1}\Box(p \wedge \neg q)$, that is, $\Box_{n-1}\Diamond(\neg p \vee q)$ holds. From $\Box_0\Diamond\neg(p \wedge q)$ using $N4$ we get $\Box_{n-1}\Box\neg(p \wedge q)$ Since, in general, in any normal modal logic $\Box$ (and, in particular, $\Box_{n-1}\Diamond$) distributes over conjunction, we obtain

$$\Box_{n-1}(\Diamond(\neg p \vee q) \wedge \Box(\neg p \vee \neg q)). \tag{4.6}$$

The schema K for $\Box$ is equivalent to $(\Box p \wedge \Diamond q) \to \Diamond(p \wedge q)$. Thus (4.6) is equivalent to $\Box_{n-1}\Diamond(\Diamond((\neg p \vee q) \wedge (\neg p \vee \neg q)))$. This in turn is equivalent to $\Box_{n-1}\Diamond\neg p$. Thus $\neg\Diamond_{n-1}\Box p$ which contradicts $\Diamond_{n-1}\Box p$.

Next, we prove $\phi \to \neg\Diamond_{n+m}\Box(p \vee q)$. Suppose ϕ holds. Then, in particular, $\neg\Diamond_n\Box p$ and $\neg\Diamond_m\Box q$ hold, and $\neg\Diamond_{n+m}\Box(p \vee q)$ is derivable by (the contrapositive of) $N7$. Therefore, $\Pi(A4)$ is a theorem in $\overline{K}_E$. □

For proving completeness of the translation Π a semantic proof suffices. The next theorem proves that for a translated formula $\Pi(\varphi)$ which is true in all $\overline{K}_E$-frames, φ is true in all $\overline{K}$-frames. If $\Pi(\varphi)$ is provable in $\overline{K}_E$ then the soundness of the $\overline{K}_E$-semantics guarantees that $\Pi(\varphi)$ is true in all $\overline{K}_E$-frames, then φ is true in all $\overline{K}$-frames and then it is provable in $\overline{K}$ (by the completeness of $\overline{K}$).

Theorem 4.3.2 For any formula φ of $\overline{K}$,

$$\text{if } \models_{\overline{K}_E} \Pi(\varphi) \text{ then } \models_{\overline{K}} \varphi.$$

Proof. Our strategy is the following. Suppose $\models_{\overline{K}_E} \Pi(\varphi)$. (i) For an arbitrary $\overline{K}$-frame $\mathcal{F}$ we construct a $\overline{K}_E$-frame $\mathcal{F}'$. $\Pi(\varphi)$ is valid in this particular frame $\mathcal{F}'$. Then we show, (ii) φ is valid in $\mathcal{F}$.

(i): Take any $\overline{K}$-frame $\mathcal{F} = (W, R)$. We construct a $\overline{K}_E$-frame $\mathcal{F}'$ as an extension of the frame $\mathcal{F}$ as follows. For any world $x \in W$ let $R''(x)$ be the R-image of x. For any finite

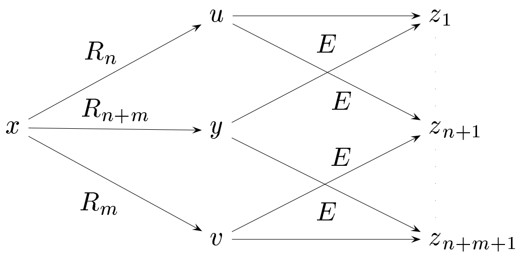


Figure 4.2: Case 1 of $N7'$.

subset Y of $R''(x)$ with $\mathrm{card}(Y) = n+1$ for n a non-negative integer, we add Y as a new world to $\mathcal{F}$. We call Y a 'set-world'. Note, every Y is non-empty. We define every relation R_m for $m \leq n$ to contain the pair (x, Y), and, we define the relation E to contain all pairs (Y, z) for $z \in Y$. Furthermore, we assume the relations R_n and E are the smallest relations satisfying these conditions. Now, define $\mathcal{F}'$ to be the relational structure

$$(W', \{R_n\}_{n \in \mathbb{N}_0}, E)$$

with W' being the set of worlds of $\mathcal{F}'$ that includes the set of worlds W of $\mathcal{F}$ and all set-worlds Y. Note, the set-worlds have no R_n-successors and the worlds in W have no E-successors. We show that $\mathcal{F}'$ is a frame for $\overline{K}_E$ by showing that $\mathcal{F}'$ satisfies the properties $N4'$–$N8'$.

$N4'$: If $R_n(x,y) \wedge E(y,z)$ holds then y must be a set-world with $\mathrm{card}(y) > n$ and $z \in y$. For $u = \{z\}$ we obtain $R_0(x,u) \wedge \forall v\ (E(u,v) \rightarrow v = z)$.

$N5'$: If $R_n(x,y)$ then y is a non-empty set-world, that is, $\exists z\ E(y,z)$ is true.

$N6'$: If $R_{n+1}(x,y)$ then y is a set-world with $\mathrm{card}(y) > n+1 > n$, that is, $R_n(x,y)$ holds as well.

Recall $N7'$:

$$\begin{aligned}&\forall xy\ R_{n+m}(x,y) \rightarrow \forall fg\ \exists uv\ (R_n(x,u) \rightarrow E(u,f(u)) \wedge R_m(x,v) \rightarrow \\ &\qquad E(v,g(v))) \rightarrow (R_n(x,u) \wedge R_m(x,v) \wedge E(y,f(u)) \wedge f(u) = g(v)).\end{aligned}$$

$R_{n+m}(x,y)$ means that y is a set-world with $\mathrm{card}(y) > n+m$. We distinguish two cases.

Case 1: $\mathrm{card}(y) = n+m+1$. Let f and g be functions mapping worlds to worlds. If there is at least one R_n-accessible set-world u with $\mathrm{card}(u) > n$ and f does not map u to one of its elements ($\neg E(u,f(u))$) or there is at least one R_m-accessible set-world v with $\mathrm{card}(v) > n$ and g does not map v to one of its elements ($\neg E(v,g(v))$) then we can choose this u or v, respectively. Then the implication

$$\begin{aligned}(4.7)\quad &(R_n(x,u) \rightarrow E(u,f(u)) \wedge R_m(x,v) \rightarrow E(v,g(v))) \rightarrow \\ &\qquad\qquad (R_n(x,u) \wedge R_m(x,v) \wedge E(y,f(u)) \wedge f(u) = g(v))\end{aligned}$$

is true because the premise is false.

Now assume, f chooses for every set-world u with $\mathrm{card}(u) > n$ some element $f(u) \in u$ and g chooses for every set-world v with $\mathrm{card}(v) > m$ some element $g(v) \in v$. The key observation for the proof is that for every set with $n+m+1$ elements every decomposition into a set u with $n+1$ elements and a set v with $m+1$ elements overlaps in at least one element. Thus, the situation is as depicted in Figure 4.2.

For finding the right u and v we follow this procedure. We start by choosing a subset $u_1 \subseteq y$ with $n+1$ elements. Suppose $f(u_1) = x_1$. If there is a subset $v \subseteq y$ with $\mathrm{card}(v) > m$

and $g(v) = x_1$ we are done. Suppose for no such subset we have $g(v) = x_1$. x_1 is marked as 'not an image of g'. Now we choose another $n+1$-element subset u_2 of y which *does not contain* x_1. Suppose $f(u_2) = x_2$. Again, if for some subset v with $\mathrm{card}(v) > m$ we find $g(v) = x_2$ we are done. If not, we mark x_2 as 'not an image of g'. We continue until we have found a suitable u and v, or until exactly $n+1$ worlds remain which are not marked 'not an image of g'. In the latter case we choose this set for u. Suppose $f(u) = x$. Take $v = y \setminus u \cup \{x\}$. $\mathrm{card}(v) = m+1$ and $g(v) \neq z$ for all $z \in y \setminus u$. Since g must select some element in v, $g(v) = x$ is the only choice. Thus, suitable u and v exist that satisfy (4.7).

Case 2: $\mathrm{card}(y) > n+m+1$. Take any subset $y' \subseteq y$ with $\mathrm{card}(y') = n+m+1$. By Case 1, we can find for any f and g subsets $u \subseteq y'$ and $v \subseteq y'$ with the property (4.7). But these are also subsets of y and therefore the property holds as well.

$N8'$: This property expresses that the union of two disjoint sets of cardinality strictly greater than n and strictly greater than m is a set with cardinality strictly greater than $n+m+1$, and this is true in $\mathcal{F}'$.

We have proved $\mathcal{F}'$ is a frame for $\overline{K}_E$.

(ii): Let $\mathcal{M} = (\mathcal{F}, \iota)$ be any model based on $\mathcal{F}$ with ι an arbitrary valuation. Define $\mathcal{M}'$ to be the model $(\mathcal{F}', \iota)$. (Observe that $\iota(p)$ does not, and need not contain set-worlds.) (ii) follows from

$$\mathcal{M}', x \models_{\overline{K}_E} \Pi(\varphi) \quad \text{iff} \quad \mathcal{M}, x \models_{\overline{K}} \varphi \tag{4.8}$$

where x is any world in W. We prove (4.8) by induction on the structure of φ. The base case in which φ is any propositional variable is trivial. The inductive step for the propositional connectives goes through easily. We consider the case φ is of the form $\blacklozenge_n\psi$. (The case for φ of the form $\blacksquare_n\psi$ is dual.) The inductive hypothesis is:

$$\mathcal{M}', x \models_{\overline{K}_E} \Pi(\psi) \quad \text{iff} \quad \mathcal{M}, x \models_{\overline{K}} \psi.$$

Suppose $\mathcal{M}', x \models_{\overline{K}_E} \Pi(\blacklozenge_n\psi)$, that is, $\Diamond_n\Box\psi$ is true at x in $\mathcal{M}'$. Then, $R_n(x, Y)$ in $\mathcal{F}'$ for some set $Y \subseteq R''(x)$ with $n+1$ elements and for all $z \in Y$ we have $\mathcal{M}', z \models_{\overline{K}_E} \psi$ and by the inductive hypothesis $\mathcal{M}, z \models_{\overline{K}} \psi$. There are at least n such z, therefore, $\mathcal{M}, x \models_{\overline{K}} \blacklozenge_n\psi$.

Conversely, suppose $\mathcal{M}, x \models_{\overline{K}} \blacklozenge_n\psi$. This means the world x has more than n successors by R in all of which ψ is true. Consequently, a set Y with cardinality $n+1$ exists that contains R-successors y of x and in all y, ψ is true. This implies, in $\mathcal{F}'$, x and Y are connected by R_n and Y is connected to all its elements by E. Thus, $\Diamond_n\Box\psi$ is true in x.

This completes the proof. □

As consequences we get the following two theorems.

Theorem 4.3.3 (Completeness of Π) The translation Π from $\overline{K}$ into $\overline{K}_E$ is complete. That is, for any formula φ of $\overline{K}$,

$$\text{if } \vdash_{\overline{K}_E} \Pi(\varphi) \text{ then } \vdash_{\overline{K}} \varphi.$$

Proof. Suppose $\Pi(\varphi)$ is a theorem in $\overline{K}_E$, that is, $\vdash_{\overline{K}_E} \Pi(\varphi)$. Then, since $\overline{K}_E$ is sound (4.4), $\models_{\overline{K}_E} \Pi(\varphi)$. By the previous theorem $\models_{\overline{K}} \varphi$. $\overline{K}$ is sound and complete. Therefore, it follows that $\vdash_{\overline{K}} \varphi$. □

Now, we can show the completeness of the semantics of $\overline{K}_E$ with respect to its axiomatisation *for translated formulae*.

Theorem 4.3.4 (Relative completeness of $\overline{\boldsymbol{K}}_E$) For any $\overline{K}$ formula φ,

$$\text{if } \models_{\overline{K}_E} \Pi(\varphi) \text{ then } \vdash_{\overline{K}_E} \Pi(\varphi).$$

Proof. If $\Pi(\varphi)$ holds in all $\overline{K}_E$-frames then φ holds in all $\overline{K}$-frames (by Theorem 4.3.2), then φ is provable in $\overline{K}$ (by the completeness of $\overline{K}$), and then $\Pi(\varphi)$ is provable in $\overline{K}_E$ (by the soundness of the translation, Theorem 4.3.1). □

4.4 Translating $\overline{\boldsymbol{K}}_E$ to first-order logic

The Theorems 4.3.1 to 4.3.4 show that inference about numerical quantification formulated as a formula φ of $\overline{K}$ can be done in the context of the multi-modal logic $\overline{K}_E$ by inference on the translation $\Pi(\varphi)$. This is not what we aim to do. We aim at using first-order resolution methods for reasoning with graded modal expressions. $\overline{K}_E$ being a multi-modal logic we can immediately use one of the three reductions to first-order logic. As one of the schemas, namely $N7$, is not first-order definable in the standard Kripke semantics, relational and functional translations are disqualified. Instead, we use the optimised functional translation and it turns out, like McKinsey's schema, $N7$ has a first-order functional approximation.

The completeness theorems for the functional translation mappings require that the modal logics are complete with respect to some class of frames. For $\overline{K}_E$ we showed only relative completeness, that is, $\overline{K}_E$ is complete for the translated $\overline{K}$ formulae only. Since this fragment of $\overline{K}_E$ is sufficient for our purpose, we have fullfilled the conditions of Theorems 3.2.2 and 3.3.2.

The functional translation for serial modalities is considerably simpler than for non-serial modalities. The accessibility relations R_n $(n \in \mathbb{N}_0)$ and E are not serial. Axiom $N5$, $\Diamond_n \Box p \to \Diamond_n \Diamond p$, specifies a weak form of seriality for E. Every world accessible by some R_n (that is, every set-world) has a successor by E.

We do not need the full expressiveness of the language of $\overline{K}_E$. A subset of formulae with characteristic patterns of modal operators $\Diamond_n$, $\Box_n$, $\Diamond$ and $\Box$ will do. For example, in the axiomatisation defining $\overline{K}_E$ the operators $\Diamond$ and $\Box$ do not occur in the scope of $\Diamond$ and $\Box$ operators, they always occur in the scope of $\Diamond_n$ and $\Box_n$ operators. This is intentional. Only these patterns make sense in our application of $\overline{K}_E$. We think of the numerical modalities picking only *sets* and the $\Diamond$ and $\Box$ operators picking only elements of these sets. For this we need a special class of formulae in which the E-successors of worlds accessible by E are irrelevant, only E-successors of worlds accessible by the relations R_n count. We are therefore permitted to assume E is serial (which is embodied the next theorem).

The language of $\overline{K}_E$ that we will use is restricted to the set of *admissible* formulae. We say a formula φ of $\overline{K}_E$ is admissible iff all $\Diamond$ and $\Box$ operators appear in the scope of a $\Diamond_n$ or $\Box_n$ operator (for some n). Examples of admissible formulae are:

$$\Diamond_n \Box p, \quad \Box_n \Diamond p, \quad \Diamond_n (p \wedge \Diamond q) \quad \text{and} \quad \Box_n(\neg \Box p \to \Diamond q).$$

The formulae $\Diamond p$ and $\Diamond_n \Box \Diamond q$ on the other hand, are not admissible. We note that the translation Π maps $\overline{K}$ formulae to admissible formulae, since the corresponding modalities for modal operators of $\overline{K}$ are $\Diamond_n \Box$ and $\Box_n \Diamond$. Evidently, any substitution instance of $N4$–$N8$ is admissible.

Theorem 4.4.1 Let φ be an admissible formula of $\overline{K}_E$. If φ is valid in a model then φ is valid in a model in which the relation E (associated with the modalities $\Diamond$ and $\Box$) is serial.

Proof. Let φ be valid in a model $\mathcal{M} = (W, \{R_n\}_{n\in\mathbb{N}_0}, E, \iota)$ Define $\mathcal{M}'$ to be a model $(W, \{R_n\}_{n\in\mathbb{N}_0}, E', \iota)$ obtained from $\mathcal{F}$ by replacing E an extension. E' includes E and all pairs (x, x) of $x \in W$ for which no $y \in W$ exists such that $E(x, y)$. Evidently, E' is serial.

We show that φ is valid in $\mathcal{M}'$. The only critical case is where φ has a formula equivalent to $\psi = \Box\phi \wedge \Box\neg\phi$ as subformula. For ψ is true at any world x in $\mathcal{M}$ iff x has no E-successor in $\mathcal{M}$. In $\mathcal{M}'$, however, ψ is false at x whenever ψ is true at x in $\mathcal{M}$ (otherwise there is an inconsistency). ψ occurs in the scope of either $\Diamond_n$ or $\Box_n$ for some n. First, we consider the case that ψ occurs in the scope of $\Diamond_n$. Let φ' be the $\Diamond_n$ subformula of φ with ψ in its scope. We may assume φ' is of the form $\Diamond_n((\psi \vee \alpha) \wedge \beta)$. Now, suppose φ' is true in a world x of $\mathcal{M}$. Then there is a $y \in W$ such that $R_n(x, y)$. Since E is weakly serial there is a $z \in W$ such that $E(y, z)$, which implies ψ is false in y of $\mathcal{M}$. Hence, φ' is true in x of $\mathcal{M}$ iff it is also true in x of $\mathcal{M}'$. Next, we consider the case that ψ occurs in the scope of $\Box_n$. Assume φ' is of the form $\Box_n((\psi \vee \alpha) \wedge \beta)$ and suppose φ' is true in $x \in W$ of $\mathcal{M}$. Then either there are or there are no y's in W such that $R_n(x, y)$. If there are y's then we argue as above. If there are no y's then φ' is trivially true in x of $\mathcal{M}'$. We conclude, φ is indeed valid in $\mathcal{M}'$. $\Box$

This theorem licenses the translation of the $\Diamond$ and $\Box$ operators without the dead-end predicate de_E. For admissible formulae in $\overline{K}_E$ the translation function π_f can be taken to be defined for the modal operators by:

$$\begin{aligned}
\pi_f(\Box_n\varphi, x) &= \neg de_n(x) \rightarrow \forall\gamma_n\ \pi_f(\varphi, [x\gamma_n]) \\
\pi_f(\Diamond_n\varphi, x) &= \neg de_n(x) \wedge \exists\gamma_n\ \pi_f(\varphi, [x\gamma_n]) \\
\pi_f(\Box\varphi, x) &= \forall\gamma_E\ \pi_f(\varphi, [x\gamma_E]) \\
\pi_f(\Diamond\varphi, x) &= \exists\gamma_E\ \pi_f(\varphi, [x\gamma_E]).
\end{aligned}$$

We are now set to compute the first-order formulae for the schemas $N4$–$N8$ of $\overline{K}_E$. This is a mechanical and tedious task, which we left to an implementation of the general translation procedure and the tool SCAN for eliminating second-order quantifiers.

In our listing of the results below, we use indexes to indicate the sorts of variables and Skolem terms. The value in the subscript of a Skolem function, say f^n_m, associates the function with the m-th modality $\Diamond_m$ and AF_m is the sort of the terms formed with f^n_m. The subscript n is part of the name of the Skolem function. It is only used to distinguish the different Skolem functions for the different instances of the clauses. (In an actual implementation, these numbers are the objects of symbolic arithmetical manipulations.)

The following presents for each schema $N4$–$N8$, (i) the functional translation, (ii) the first-order equivalent formulation and (iii) its clause form.

$N4$: $\Box_0\Diamond p \rightarrow \Box_n\Box p$.

The functional translation $\Pi_f(N4)$ is

$$\forall P\ \forall x\ ((\neg de_0(x) \rightarrow \forall\alpha_0\exists\beta_E P[x\alpha_0\beta_E]) \rightarrow (\neg de_n(x) \rightarrow \forall\gamma_n\ \forall\delta_E\ P[x\gamma\delta])).$$

This has a first-order equivalent formulation, namely

$$\begin{aligned}
&\forall x\ de_0(x) \rightarrow de_n(x)\ \wedge \\
&\forall x\ (\neg de_n(x) \rightarrow (\forall\gamma_n\ \forall\delta_E\ \exists\alpha_0\ \forall\beta_E\ [x\gamma_n\delta_E] = [x\alpha_0\beta_E])).
\end{aligned} \tag{4.9}$$

The clause form is:

$$\neg de_0(x) \vee de_n(x).$$
$$de_n(x) \vee [x\gamma_n\delta_E] = [xf_0^n(x,\gamma_n,\delta_E)\beta_E].$$

$N5$: Any instance of this schema is valid in every frame in which E is a serial relation.

$N6$: $\Box_n p \to \Box_{n+1} p$.

$\Pi_f(N6)$ is given by

$$\forall P\forall x\ ((\neg de_n(x) \to \forall\alpha_n\ P[x\alpha_n]) \to (\neg de_{n+1}(x) \to \forall\beta_{n+1}\ P[x\beta_{n+1}])),$$

the first-order equivalent by:

$$\begin{aligned}&\forall x\ (de_n(x) \to de_{n+1}(x)) \wedge \\ &\forall x\ (\neg de_{n+1}(x) \to \forall\beta_{n+1}\ \exists\alpha_n\ [x\beta_{n+1}] = [x\alpha_n]),\end{aligned} \tag{4.10}$$

and the clause form by:

$$\neg de_n(x) \vee de_{n+1}(x).$$
$$de_{n+1}(x) \vee [x\beta_{n+1}] = [xg_n^n(x,\beta_{n+1})].$$

$N7$: $\Diamond_{n+m}\Box(p \vee q) \to (\Diamond_n\Box p \vee \Diamond_m\Box q)$

is translated by Π_f to:

$$\begin{aligned}&\forall PQ\ \forall x\ ((\neg de_{n+m}(x) \wedge \exists\alpha_{n+m}\ \forall\beta_E\ (P[x\alpha_{n+m}\beta_E] \vee Q[x\alpha_{n+m}\beta_E])) \\ &\qquad \to ((\neg de_n(x) \wedge \exists\alpha_n\ \forall\gamma_E\ P[x\alpha_n\gamma_E]) \vee (\neg de_m(x) \wedge \exists\alpha_m\ \forall\delta_E\ Q[x\alpha_m\delta_E]))).\end{aligned}$$

This is equivalent to the formula

$$\begin{aligned}&\forall x\ (de_{n+m}(x) \vee (\neg de_n(x) \wedge \neg de_m(x) \wedge \forall\alpha_{n+m}\ \forall\gamma_E\ \forall\delta_E\ \exists\beta_E\ \exists\alpha_n\ \exists\alpha_m \\ &\qquad ([x\alpha_{n+m}\beta_E] = [x\alpha_n\gamma_E(\alpha_n)] \wedge [x\alpha_{n+m}\beta_E] = [x\alpha_n\delta_E(\alpha_m)])))).\end{aligned}$$

This formula is second-order, for note the terms $\gamma_E(\alpha_n)$ and $\delta_E(\alpha_m)$. We get a first-order equivalent formula if we apply the quantifier exchange rule Υ to $\Pi_f(N7)$:

$$\begin{aligned}&\forall PQ\ \forall x\ ((\neg de_{n+m}(x) \wedge \exists\alpha_{n+m}\ \forall\beta_E\ (P[x\alpha_{n+m}\beta_E] \vee Q[x\alpha_{n+m}\beta_E])) \to \\ &\qquad ((\neg de_n(x) \wedge \exists\alpha_n\ \forall\gamma_E\ P[x\alpha_n\gamma_E]) \vee (\neg de_m(x) \wedge \forall\delta_E\ \exists\alpha_m\ Q[x\alpha_m\delta_E])))\end{aligned}$$

which is equivalent to the first-order formula

$$\begin{aligned}&\forall x\ (de_{n+m}(x) \vee (\neg de_n(x) \wedge \neg de_m(x) \wedge \\ &\quad \forall\alpha_{n+m}\ \forall\gamma_E\ \exists\beta_E\ \exists\alpha_m\ \forall\delta_E\ \exists\alpha_n\ ([x\alpha_{n+m}\beta_E] = [x\alpha_n\gamma_E] \vee [x\alpha_{n+m}\beta_E] = [x\alpha_m\delta_E]))).\end{aligned}$$

The clause form is:

$$de_{n+m}(x) \vee \neg de_n(x).$$
$$de_{n+m}(x) \vee \neg de_m(x).$$
$$de_{n+m}(x) \vee [x\alpha_{n+m}h1_E^{nm}(x,\alpha_{n+m},\gamma_E)] = [xh2_n^{nm}(x,\alpha_{n+m},\gamma_E,\delta_E)\gamma_E].$$
$$de_{n+m}(x) \vee [x\alpha_{n+m}h1_E^{nm}(x,\alpha_{n+m},\gamma_E)] = [xh3_m^{nm}(x,\alpha_{n+m},\gamma_E)\delta_E].$$

$N8$: $(\Diamond_n\Box(p \wedge q) \wedge \Diamond_m\Box(p \wedge \neg q)) \rightarrow \Diamond_{n+m+1}\Box p$

translates to $\Pi_f(N8)$:

$$\begin{aligned}\forall PQ\ \forall x\ (((\neg de_n(x) \wedge \exists\alpha_n\ \forall\beta_E\ (P[x\alpha_n\beta_E] \wedge Q[x\alpha_n\beta_E])) \wedge \\ (\neg de_m(x) \wedge \exists\alpha_m\ \forall\gamma_E\ (P[x\alpha_m\gamma_E] \wedge \neg Q[x\alpha_m\gamma_E]))) \rightarrow \\ (\neg de_{n+m+1}(x) \wedge \exists\alpha_{n+m+1}\ \forall\delta_E\ P[x\alpha_{n+m}\delta_E]))\end{aligned}$$

This is equivalent to

$$\begin{aligned}&\forall x\ (de_n(x) \vee de_m(x) \vee \forall\alpha_n\ \forall\alpha_m\ \exists\alpha_{n+m+1}\ \forall\delta_E\ (\neg de_{n+m+1}(x) \vee \\ &\quad \exists\beta_E\ \exists\gamma_E\ [x\alpha_n\beta_E] = [x\alpha_m\gamma_E]) \wedge (\exists\beta_E\ \exists\gamma_E\ [x\alpha_n\beta_E] = [x\alpha_m\gamma_E] \vee \\ &\quad \exists\beta_E\ [x\alpha_n\beta_E] = [x\alpha_{n+m+1}\delta_E] \vee \exists\gamma_E\ [x\alpha_m\gamma_E] = [x\alpha_{n+m+1}\delta_E])).\end{aligned}$$

The clause form is

$$\begin{aligned}&de_n(x) \vee de_m(x) \vee \neg de_{n+m+1}(x) \vee \\ &\qquad [x\alpha_n k1_E^{nm}(x,\alpha_n,\alpha_m)] = [x\alpha_m k2_E^{nm}(x,\alpha_n,\alpha_m)]. \\ &de_n(x) \vee de_m(x) \vee \\ &\qquad [x\alpha_n k3_E^{nm}(x,\alpha_n,\alpha_m)] = [x\alpha_m k4_E^{nm}(x,\alpha_n,\alpha_m)] \vee \\ &\qquad [x\alpha_n k5_E^{nm}(x,\alpha_n,\delta_E)] = [xk_{n+m+1}^{nm}(x,\alpha_n,\alpha_m)\delta_E] \vee \\ &\qquad [x\alpha_m k6_E^{nm}(x,\alpha_m,\delta_E)] = [xk_{n+m+1}^{nm}(x,\alpha_n,\alpha_m)\delta_E].\end{aligned}$$

The clauses sets of the translation of $\overline{K}_E$ are schemas. They represent the conjunction of all clause instances with the n and m taking on concrete non-negative integer values. This can be exploited in certain generalisations. For example, the subformula $\forall x\ de_n(x) \rightarrow de_{n+1}(x)$ of (4.10) can be generalised to

$$\forall x\ (de_n(x) \rightarrow de_m(x)) \qquad \text{for all } m > n.$$

This formula subsumes the subformula $\forall x\ (de_0(x) \rightarrow de_n(x))$ of the translation (4.9) of $N4$. The remaining part of (4.10) can also be generalised to:

$$\forall x\ (\neg de_m(x) \rightarrow (\forall\beta_m\ \exists\alpha_n\ [x\beta_m] = [x\alpha_n])) \qquad \text{for all } m > n.$$

The clause form is

$$de_m(x) \vee [x\beta_m] = [xg_n^{nm}(x,\beta_m)] \qquad \text{for all } m > n. \tag{4.11}$$

Recall the relational translation of $N6$. We noted that $N6'$ generalises to

$$R_m \subseteq R_n \qquad \text{for all } m \geq n.$$

This ordering on the accessibility relations $\{R_n\}_{n\in\mathbb{N}_0}$ induces a linear ordering on the set $\{AF_n\}_{n\in\mathbb{N}_0}$ of sets of accessibility functions. We capture this ordering by the subsort declaration

$$AF_m \sqsubseteq AF_n \qquad \text{for all } m \geq n. \tag{4.12}$$

In a resolution calculus this declaration has the same effect as clause (4.11). We therefore replace (4.11) by the subsort declaration (4.12).

The axiomatisation of $\overline{K}_E$ reduces to the following set first-order clauses which defines the theory for $\overline{K}_E$.

$P1$	$de_n(x) \vee [x\alpha_n\beta] = [xf_0^n(x,\alpha_n,\beta)\gamma]$
$P2$	$AF_m \sqsubseteq AF_n$ for all $m > n$
$P3$	$\neg de_n(x) \vee de_m(x)$ for all $m > n$
$P4$	$de_{n+m}(x) \vee [x\alpha_{n+m}h1^{nm}(x,\alpha_{n+m},\beta)] = [xh2_n^{nm}(x,\alpha_{n+m},\beta,\gamma)\beta]$
$P5$	$de_{n+m}(x) \vee [x\alpha_{n+m}h1^{nm}(x,\alpha_{n+m},\beta)] = [xh3_m^{nm}(x,\alpha_{n+m},\beta)\gamma]$
$P6$	$de_{\max(n,m)}(x) \vee \neg de_{n+m+1}(x) \vee [x\alpha_n k1^{nm}(x,\alpha_n,\beta_m)] = [x\beta_m k2^{nm}(x,\alpha_n,\beta_m)]$
$P7$	$de_{\max(n,m)}(x) \vee [x\alpha_n k3^{nm}(x,\alpha_n,\beta_m)] = [x\beta_m k4^{nm}(x,\alpha_n,\beta_m)] \vee$ $[x\alpha_n k5^{nm}(x,\alpha_n,\gamma)] = [xk_{n+m+1}^{nm}(x,\alpha_n,\beta_m)\gamma] \vee$ $[x\beta_m k6^{nm}(x,\beta_m,\gamma)] = [xk_{n+m+1}^{nm}(x,\alpha_n,\beta_m)\gamma]$.

(The variables β and γ and the functions $h1^{nm}$ and $k1^{nm}$–$k6^{nm}$ without index are variables and functions of sort AF_E.)

Theorem 4.4.2 For any $\overline{K}$-formula φ,

$$\varphi \text{ is a } \overline{K}\text{-theorem} \quad \text{iff} \quad (P1\text{–}P7) \rightarrow \Upsilon\Pi_f(\varphi) \text{ is a first-order theorem.}$$

Proof. By Theorem 3.3.2. □

By way of examples we illustrate how $P1$–$P7$ are used during inference with a resolution-based theorem prover. B. Nebel provided the following examples.

Example 4.4.3 The set $A = \{\blacklozenge_0\blacklozenge_3\top,\ \blacklozenge_0\blacksquare_3\bot,\ \blacksquare_1\bot\}$ is an inconsistent set of $\overline{K}$-formulae. With theory resolution we can show the inconsistency in a single step. The $\overline{K}_E$ and the first-order formulations of $\blacklozenge_0\blacklozenge_3\top$, $\blacklozenge_0\blacksquare_3\bot$ and $\blacksquare_1\bot$ are respectively

$$\begin{array}{lll} \Diamond_0\Box\Diamond_3\Box\top & \text{and} & \neg de_0[] \wedge \neg de_3[\underline{\alpha}_0\beta], \\ \Diamond_0\Box\Box_3\Diamond\bot & \text{and} & \neg de_0[] \wedge de_3[\underline{\gamma}_0\delta], \\ \Box_1\Diamond\bot & \text{and} & de_1[]. \end{array}$$

The set A is represented by the following set of clauses:

1. $\neg de_0[]$	3. $de_3[\underline{\gamma}_0\delta]$
2. $\neg de_3[\underline{\alpha}_0\beta]$	4. $de_1[]$,

where β and δ are variables and $\underline{\alpha}_0$ and $\underline{\gamma}_0$ are Skolem constants. Letting $n = 0$ and $m = 0$ in $P6$ and using the substitution

$$\{\alpha_0' \mapsto \underline{\alpha}_0,\ \ \beta_0' \mapsto \underline{\gamma}_0,\ \ \beta \mapsto k1^{00}([],\underline{\alpha}_0,\underline{\gamma}_0),\ \ \delta \mapsto k2^{00}([],\underline{\alpha}_0,\underline{\gamma}_0)\}$$

$P6$ simultaneously resolves with 1.–4. yielding the empty clause.

The next example is more complicated.

Example 4.4.4 The set $B = \{\blacklozenge_0\blacklozenge_0\blacklozenge_3\top,\ \ \blacklozenge_0\blacklozenge_0\blacksquare_3\bot,\ \ \blacklozenge_0\blacksquare_1\bot,\ \ \blacksquare_1\bot\}$ of $\overline{K}$-formulae is also inconsistent. B contains the formulae of A (from Example 4.4.3) prefixed with a $\blacklozenge_0$ and the formula $\blacksquare_1\bot$. The set of expressions in B is represented by the following clauses.

1. $\neg de_0[]$	5. $de_3[\underline{\gamma}_0\alpha'\underline{\delta}_0\beta']$
2. $\neg de_0[\underline{\alpha}_0\alpha]$	6. $de_1[\underline{\varepsilon}_0\alpha'']$
3. $\neg de_3[\underline{\alpha}_0\alpha\underline{\beta}_0\beta]$	7. $de_1[]$
4. $\neg de_0[\underline{\gamma}_0\alpha']$	

For the refutation we use $P1$ with $n = 0$ and $P6$ with $n = 0$ and $m = 0$. $P1$ can immediately be simplified with clause 1. The instances are:

$P1'$ $[x f_0^0([], \alpha_0, \beta)\gamma] = [x\alpha_0\beta]$

$P6'$ $de_0(x) \vee \neg de_1(x) \vee [x\alpha_0 k1^{00}(x, \alpha_0, \beta_0)] = [x\beta_0 k2^{00}(x, \alpha_0, \beta_0)]$.

The result of simultaneously resolving $P6'$, 1. and 7. with unifier $\{x \mapsto []\}$ is

8. $[\alpha_0 k1^{00}([], \alpha_0, \beta_0)] = [\beta_0 k2^{00}([], \alpha_0, \beta_0)]$.

Paramodulating with 8. and with unifier $\{\alpha_0 \mapsto \underline{\alpha}_0,\ \ \alpha \mapsto k1^{00}([], \underline{\alpha}_0, \beta_0)\}$, 3. becomes (this means we do equality replacement with unification in 3. using the equation 8.)

9. $\neg de_3([\beta_0 k2^{00}([], \underline{\alpha}_0, \beta_0)\underline{\beta}_0\beta])$.

This becomes

10. $\neg de_3[\alpha_0\beta'\underline{\beta}_0\beta]$

when paramodulating with $P1'$ using the unifier

$$\{\beta_0 \mapsto f_0^0([], \alpha_0, \beta'),\ \ \gamma \mapsto k2^{00}([], \underline{\alpha}_0, \beta_0)\}.$$

We resolve $P6'$ and 4. using unifier $\{x \mapsto [\underline{\gamma}_0\alpha],\ \ \alpha' \mapsto \alpha\}$ to get

11. $\neg de_1[\underline{\gamma}_0\alpha] \vee [\underline{\gamma}_0\alpha\alpha_0' k1^{00}([\underline{\gamma}_0\alpha], \alpha_0', \beta_0')] = [\underline{\gamma}_0\alpha\beta_0' k2^{00}([\underline{\gamma}_0\alpha], \alpha_0', \beta_0')]$.

Now, use the unifier

$$\begin{gathered}\{\alpha_0 \mapsto \underline{\gamma}_0,\ \ \alpha' \mapsto \beta' \mapsto \alpha,\ \ \alpha_0' \mapsto \underline{\delta}_0,\ \ \beta_0' \mapsto \underline{\beta}_0,\\ \beta' \mapsto k1^{00}([\underline{\gamma}_0\alpha], \underline{\delta}_0, \underline{\beta}_0),\ \ \beta \mapsto k2^{00}([\underline{\gamma}_0\alpha], \underline{\delta}_0, \underline{\beta}_0)\}\end{gathered}$$

and apply E-resolution to 5., 10. and 11. and get

12. $\neg de_1[\underline{\gamma}_0\alpha]$.

(This means we resolve between 5. and 10. using an equation in 11.) Resolving this with 6. using E-resolution with 8. yields the empty clause. The unifier is

$$\{\alpha_0 \mapsto \underline{\epsilon}_0,\ \ \alpha'' \mapsto k1^{00}([], \underline{\epsilon}_0\underline{\gamma}_0),\ \ \beta_0 \mapsto \underline{\gamma}_0,\ \ \alpha'' \mapsto k2^{00}([], \underline{\epsilon}_0\underline{\gamma}_0)\}.$$

Example 4.4.5 In this example we show

$$(\blacklozenge_n p \wedge \blacklozenge_m q \wedge \blacksquare_0 \neg(p \wedge q)) \to \blacklozenge_{n+m+1}(p \vee q) \tag{4.13}$$

is a theorem in $\overline{K}$ by showing that the following set of clauses is refutable. The set represents the negation of the theorem.

1. $\neg de_n[]$	5. $de_0[] \vee \neg P[\beta_0\underline{\gamma}] \vee \neg Q[\beta_0\underline{\gamma}]$
2. $P[\underline{\alpha}_n\alpha]$	6. $de_{n+m+1}[] \vee \neg P[\alpha_{n+m+1}\underline{\delta}]$
3. $\neg de_m[]$	7. $de_{n+m+1}[] \vee \neg Q[\alpha_{n+m+1}\underline{\delta}]$
4. $Q[\underline{\beta}_m\alpha']$	

5. can be resolved with $P3$, letting $n = 0$ and $m = n$, and 1. yielding

5′. $\neg P[\beta_0\underline{\gamma}] \vee \neg Q[\beta_0\underline{\gamma}]$.

This can be paramodulated using the equation in $P1$ and using the unifier

$$\{x \mapsto [],\ \ \beta_0 \mapsto f_0^n([], \alpha_n, \beta),\ \ \gamma \mapsto \underline{\gamma}\}.$$

The result is:

$5''.\quad \neg P[\alpha_n \beta] \vee \neg Q[\alpha_n \beta].$

Resolve this with 2. and get

$8.\quad \neg Q[\underline{\alpha}_n \beta].$

Now, take 6. and paramodulate with $P7$ using the second equation and the unifier

$$\{x \mapsto [],\ \ \alpha_{n+m+1} \mapsto k_{n+m+1}^{nm}([], \alpha_n, \beta_m),\ \ \beta \mapsto \underline{\delta}\}$$

and obtain

$9.\quad de_{n+m+1}[] \vee \neg P([\alpha_n k5^{nm}([], \alpha_n, \underline{\delta})]) \vee de_{\max(n,m)}[] \vee$
$[\alpha_n k3^{nm}([], \alpha_n, \beta_m)] = [\beta_m k4^{nm}([], \alpha_n, \beta_m)] \vee$
$[\beta_m k6^{nm}([], \beta_m, \underline{\delta})] = [k_{n+m+1}^{nm}([], \alpha_n, \beta_m)\underline{\delta}].$

The $de_{\max(n,m)}[]$ literal can be eliminated from 9. with either 1. or 3. in one resolution step. The clause

$9'.\quad de_{n+m+1}[] \vee \neg P([\alpha_n k5^{nm}([], \alpha_n, \underline{\delta})]) \vee$
$[\alpha_n k3^{nm}([], \alpha_n, \beta_m)] = [\beta_m k4^{nm}([], \alpha_n, \beta_m)] \vee$
$[\beta_m k6^{nm}([], \beta_m, \underline{\delta})] = [k_{n+m+1}^{nm}([], \alpha_n, \beta_m)\underline{\delta}]$

remains. Take 7. and paramodulate with $9'$. and unifier

$$\{\alpha_{n+m+1} \mapsto k_{n+m+1}^{nm}([], \alpha_n, \beta_m)\}.$$

We obtain

$10.\quad de_{n+m+1}[] \vee \neg P([\alpha_n k5^{nm}([], \alpha_n, \underline{\delta})]) \vee \neg Q([\beta_m k6^{nm}([], \beta_m, \underline{\delta})]) \vee$
$[\alpha_n k3^{nm}([], \alpha_n, \beta_m)] = [\beta_m k4^{nm}([], \alpha_n, \beta_m)].$

Use 2. and 4. to get rid of the $\neg P$ and the $\neg Q$ literals. The unifier is

$$\{\alpha_n \mapsto \underline{\alpha}_n,\ \ \beta_m \mapsto \underline{\beta}_m,\ \ \alpha \mapsto k5^{nm}([], \underline{\alpha}_n, \underline{\delta}),\ \ \alpha' \mapsto k6^{nm}([], \underline{\beta}_m, \underline{\delta})\}.$$

10. becomes

$11.\quad de_{n+m+1}[] \vee [\underline{\alpha}_n k3^{nm}([], \underline{\alpha}_n, \underline{\beta}_m)] = [\underline{\beta}_m k4^{nm}([], \underline{\alpha}_n, \underline{\beta}_m)].$

This we can use to paramodulate with 8. The unifier is

$$\{\beta \mapsto k3^{nm}([], \underline{\alpha}_n, \underline{\beta}_m)\}$$

and the result is:

$12.\quad de_{n+m+1}[] \vee \neg Q([\underline{\beta}_m k4^{nm}([], \underline{\alpha}_n, \underline{\beta}_m)]).$

Resolve this with 4. which yields

$13.\quad de_{n+m+1}[].$

Now we use $P6$ and get

14. $de_{\max(n,m)}[] \vee [\alpha_n k1^{nm}([], \alpha_n, \beta_m)] = [\beta_m k2^{nm}([], \alpha_n, \beta_m)]$.

Get rid of the $de_{\max(n,m)}[]$ literal by resolving with either 1. or 3. The equation

14′. $[\alpha_n k1^{nm}([], \alpha_n, \beta_m)] = [\beta_m k2^{nm}([], \alpha_n, \beta_m)]$

remains. We use 8. again and paramodulate with 14′. substituting with

$$\{\alpha_n \mapsto \underline{\alpha}_n, \;\; \beta \mapsto k1^{nm}([], \alpha_n, \beta_m)\}$$

which leaves

15. $\neg Q([\beta_m k2^{nm}([], \underline{\alpha}_n, \beta_m)])$.

In the last step we resolve 15. and 4. with unifier

$$\{\beta_m \mapsto \underline{\beta}_m, \;\; \alpha \mapsto k2^{nm}([], \alpha_n, \beta_m)\}$$

to get the empty clause.

In the functional translation we can prove instances of formulae with concrete values assigned to the n and m in the modal operators. There are examples of formulae for which the proofs with symbolic arithmetic terms instead of concrete values work as well. However, this approach may not always work. The formula (4.14) below provides an example of a theorem which is true for all n and m (that satisfy the required restriction), but which can be proved in our system only for concrete instances of n and m. The situation may be worse. It may be the case that the proof of a formula for a particular concrete instance n depends on the instance of the formula for $n-1$, and the proof of this instance depends on the formula for $n-2$, and so on, to the formula for 0. Call this 'induction on foot'. We now demonstrate a process of how a schema like (4.14) can be proved in our system by (ordinary) induction for all values followed by a translation step which yields a lemma we add to our theory.

Suppose there are at least twenty objects in p and at least twenty objects in q and in all thirty objects exist. Then we expect the intersection of p and q to contain at least 10 objects. Our intuition is captured by the following formula

$$\blacklozenge_n p \wedge \blacklozenge_m q \wedge \blacksquare_j \neg(p \wedge q) \rightarrow \blacklozenge_{n+m+1-j}(p \vee q) \tag{4.14}$$

with $n+m+1-j \geq 0$, if we let $n = m = 19$ and $j = 9$. In Example 4.4.5 we showed (4.14) for the case that $j = 0$. Unfortunately there is no resolution-based proof for the general case. In the next theorem we use induction to prove (4.14).

Theorem 4.4.6 (4.14) is a theorem in $\overline{K}$.

Proof. The proof is by induction on j. We proved the base case in Example 4.4.5. Let $j > 0$. As induction hypothesis assume

$$\blacklozenge_n p \wedge \blacklozenge_m q \wedge \blacksquare_{j-1} \neg(p \wedge q) \rightarrow \blacklozenge_{n+m+1-(j-1)}(p \vee q)$$

holds. Assume further $\blacklozenge_n p$, $\blacklozenge_m q$, and $\blacksquare_j \neg(p \wedge q)$ hold. That $\blacksquare_j \neg(p \wedge q)$ holds implies $(\neg\blacksquare_{j-1}\neg(p \wedge q) \wedge \blacksquare_j \neg(p \wedge q)) \vee \blacksquare_{j-1}\neg(p \wedge q)$ holds.

Suppose $\blacksquare_{j-1}\neg(p \wedge q)$ holds. Apply the induction hypothesis to get $\blacklozenge_{n+m+1-(j-1)}(p \vee q)$, which implies $\blacklozenge_{n+m+1-j}(p \vee q)$ holds, by using $A2$.

For the second case assume $\neg\blacksquare_{j-1}\neg(p \wedge q) \leftrightarrow \blacklozenge_{j-1}(p \wedge q)$ holds. Let $k = n + m$, $p = p \wedge \neg q$ and $q = p \wedge q$ in $A10$. Then

$$\blacklozenge_k p \wedge \blacksquare_k \neg(p \wedge q) \rightarrow \blacklozenge_{k-n}(p \wedge \neg q).$$

From $\blacklozenge_n p$, respectively $\blacklozenge_m q$, and $\blacksquare_j \neg(p \wedge q)$ we infer that $\blacklozenge_{n-j}(p \wedge \neg q)$, respectively $\blacklozenge_{m-j}(\neg p \wedge q)$, holds. Hence, by $A12$,

$$\blacklozenge_{m+n+1-j-1}((p \wedge \neg q) \vee (\neg p \wedge q))$$

holds. Using $A12$ again, this time applied to the formulae

$$\blacklozenge_{m+n+1-j-1}((p \wedge \neg q) \vee (\neg p \wedge q)) \quad \text{and} \quad \blacklozenge_{j-1}(p \wedge q),$$

we conclude $\blacklozenge_{m+n+1-j}(p \vee q)$ holds. This proves the theorem. □

The next result shows we can replace the schema $N8$ in $\overline{K}_E$ by the corresponding $\overline{K}_E$-formulation of the formula (4.14).

Theorem 4.4.7 $N8$ of $\overline{K}_E$ can be replaced by

$$\Diamond_n \Box p \wedge \Diamond_m \Box q \wedge \Box_j \Diamond \neg(p \wedge q) \rightarrow \Diamond_{n+m+1-j} \Box(p \vee q) \tag{4.15}$$

for $n + m + 1 - j \geq 0$.

Proof. In Theorem 4.4.6 we proved its $\overline{K}$-formulation (4.14) is a theorem. Thus, by Theorem 4.3.1, (4.15) holds in $\overline{K}_E$. It remains to show (4.15) implies $N8$. This is immediate if we let $j = 0$ and substitute $p \wedge q$ for p and $p \wedge \neg q$ for q exploiting $\Box_0 \Diamond \top \leftrightarrow \top$ ($N4$). □

Although replacing $N8$ with (4.15) does not increase the number of provable formulae, we avoid the induction argument necessary for proving (4.15) which we would have to provide by hand as we do not have an induction theorem prover at our disposal. Also, we avoid proving instances of (4.15).

The functional translation of (4.15) into first-order logic is somewhat more complicated than that of $N8$. It is given by

$$\begin{aligned} \forall PQ\ \forall x\ (((\neg de_n(x) \wedge \exists \alpha_n\ \forall \beta_E\ P[x\alpha_n\beta_E]) \wedge (\neg de_m(x) \wedge \exists \alpha_m\ \forall \beta_E\ Q[x\alpha_m\beta_E]) \wedge \\ (\neg de_j(x) \rightarrow \forall \alpha_j\ \exists \beta_E\ \neg(P[x\alpha_j\beta_E] \wedge Q[x\alpha_j\beta_E]))) \rightarrow ((\neg de_{n+m+1-j}(x) \wedge \\ \exists \alpha_{n+m+1\text{-}j}\ \forall \beta_E\ (P[x\alpha_{n+m+1-j}\beta_E] \vee Q[x\alpha_{n+m+1-j}\beta_E])))). \end{aligned}$$

Like $\Pi_f(N7)$ this formula cannot be reduced to a first-order formula. We swap the quantifiers $\exists \gamma_{n+m+1\text{-}j}$ and $\forall \delta_E$. The quantification elimination algorithm SCAN produces then for this input the following clauses:

$P8$ $\quad de_{\max(n,m)}(x) \vee \neg de_{n+m+1-j}(x) \vee$
$\quad [xf7_j^{nmj}(x, \alpha_n, \alpha_m)\beta] = [x\alpha_n f5^{nmj}(x, \alpha_n, \beta)]$

$P9$ $\quad de_{\max(n,m)}(x) \vee \neg de_{n+m+1-j}(x) \vee$
$\quad [xf7_j^{nmj}(x, \alpha_n, \alpha_m)\beta] = [x\alpha_m f6^{nmj}(x, \alpha_m, \beta)]$

$$\begin{array}{ll} P10 & de_{\max(n,m)}(x) \vee \neg de_j(x) \vee \\ & \quad [xf1^{nmj}_{n+m+1-j}(x,\gamma,\alpha_n)\gamma] = [x\alpha_n f2^{nmj}(x,\gamma,\alpha_n)] \vee \\ & \quad [xf3^{nmj}_{n+m+1-j}(x,\gamma,\alpha_m)\gamma] = [x\alpha_m f4^{nmj}(x,\gamma,\alpha_m)] \\ P11 & de_{\max(n,m)}(x) \vee \\ & \quad [xf1^{nmj}_{n+m+1-j}(x,\gamma,\alpha_n)\gamma] = [x\alpha_n f2^{nmj}(x,\gamma,\alpha_n)] \vee \\ & \quad [xf3^{nmj}_{n+m+1-j}(x,\gamma,\alpha_m)\gamma] = [x\alpha_m f4^{nmj}(x,\gamma,\alpha_m)] \vee \\ & \quad [xf7^{nmj}_j(x,\alpha_n,\alpha_m)\beta] = [x\alpha_n f5^{nmj}(x,\alpha_n,\beta)] \\ P12 & de_{\max(n,m)}(x) \vee \\ & \quad [xf1^{nmj}_{n+m+1-j}(x,\gamma,\alpha_n)\gamma] = [x\alpha_n f2^{nmj}(x,\gamma,\alpha_n)] \vee \\ & \quad [xf3^{nmj}_{n+m+1-j}(x,\gamma,\alpha_m)\gamma] = [x\alpha_m f4^{nmj}(x,\gamma,\alpha_m)] \vee \\ & \quad [xf7^{nmj}_j(x,\alpha_n,\alpha_m)\beta] = [x\alpha_m f6^{nmj}(x,\alpha_m,\beta)] \end{array}$$

together with the clause

$$de_n(x) \vee de_m(x) \vee \neg de_j(x) \vee \neg de_{n+m+1-j}(x), \tag{4.16}$$

which is implicit in $P1$–$P7$. We can show that, for any positive integers n, m and j,

$$\exists k\ l\ (k,l) \in \{n,m\} \times \{j, n{+}m{+}1{-}j\} \quad \text{such that} \quad k \geq l.$$

For, suppose not. Suppose n, m and j exist such that for any k and l with $(k,l) \in \{n,m\} \times \{j, n{+}m{+}1{-}j\}$ we have $k < l$. Then, $n < j$, $m < j$ and $n < n+m+1-j$. Hence, $j < m+1$, and thus, $m < j < m+1$, which cannot be for j a positive integer.

If the values n, m and j are such that we can choose k and l with k strictly larger than l then (4.16) is subsumed by $P3$. Otherwise, if the values are such that we can choose identical k and l, then (4.16) is valid in every frame. In either case (4.16) is redundant.

We conclude this section with an example (supplied to us by W. Nutt) in which we exhibit the computational effect of using the clauses $P8$–$P12$.

Example 4.4.8 Suppose the universe consists of at most thirty objects. If there are at least twenty objects in p and there are at least twenty objects in q, then there are at least ten objects in $p \wedge q$. A standard tableaux system for the number operators would generate twenty witnesses for p, twenty witnesses for q and then it would need to identify ten of them in order not to exceed the limit of thirty. But there are combinatorially many ways for identifying ten of them.

In our system we prove the statement by showing the following set of $\overline{K}$-formulae is inconsistent:

$$\{\blacklozenge_{19} p,\ \blacklozenge_{19} q,\ \blacksquare_{30}\bot,\ \blacksquare_9 \neg(p \wedge q)\}.$$

We can choose any other suitable combination of numbers. This will not change the structure of the proof at all. The functional translations are:

$$\{\neg de_{19}[] \wedge P[a_{19}\alpha] \vee \neg de_{19}[] \wedge Q[b_{19}\alpha] \vee de_{30}[] \vee de_9[] \vee \neg P[\beta_9 c] \vee \neg Q[\beta_9 c]\}.$$

The corresponding set of clauses consists of:

1.	$\neg de_{19}[]$	4.	$de_{30}[]$
2.	$P[\underline{\alpha}_{19}\alpha]$	5.	$de_9[] \vee \neg P[\beta_9\underline{\gamma}] \vee \neg Q[\beta_9\underline{\gamma}]$
3.	$Q[\underline{\beta}_{19}\beta]$		

Resolve 5. with $P3$ and 1. and eliminate the $de_9[]$ literal from 5. leaving:

$5'$. $\neg P[\beta_9 \underline{\gamma}] \vee \neg Q[\beta_9 \underline{\gamma}]$

We resolve the instance of $P9$ with $n = m = 19$, $j = 9$, namely

$$P9' \quad de_{19}[] \vee \neg de_{30}[] \vee [f7_9^{19\,19\,9}([], \alpha_{19}, \alpha'_{19})\beta] = [\alpha_{19} f6^{19\,19\,9}([], \alpha'_{19}, \beta)],$$

with 1. and 4. and obtain

6. $[f7_9^{19\,19\,9}([], \alpha_{19}, \alpha'_{19})\beta] = [\alpha'_{19} f6^{19\,19\,9}([], \alpha'_{19}, \beta)]$.

Applying the unifier $\{\beta_9 \mapsto f7_9^{19\,19\,9}([], \alpha_{19}, \alpha'_{19}),\ \ u \mapsto \underline{\gamma}\}$, we can use this in $\underline{\alpha}$ paramodulation step with $5'$. resulting in

7. $\neg P[\alpha'_{19} f6^{19\,19\,9}([], \alpha'_{19}, \underline{\gamma})] \vee \neg Q[f7_9^{19\,19\,9}([], \alpha_{19}, \alpha'_{19})\underline{\gamma}]$

Unify in 2. and 7. with $\{\alpha'_{19} \mapsto \underline{\alpha}_{19}, \alpha \mapsto f6^{19\,19\,9}([], \alpha'_{19}, \underline{\gamma})\}$. Resolving 2. and 7. yields

8. $\neg Q[f7_9^{19\,19\,9}([], \alpha_{19}, \underline{\alpha}_{19})\underline{\gamma}]$.

Now we use the following instance of $P8$:

$$P8' \quad de_{19}(x) \vee \neg de_{30}(x) \vee [x f7_9^{19\,19\,9}(x, \alpha_{19}, \alpha'_{19})\beta] = [x\alpha_{19} f5^{19\,19\,9}(x, \alpha_{19}, \beta)]$$

This can be reduced with 1. and 4. to the equation

9. $[f7_9^{19\,19\,9}([], \alpha_{19}, \alpha'_{19})\beta] = [\alpha_{19} f5^{19\,19\,9}([], \alpha_{19}, \beta)]$,

which we can now use in $\underline{\alpha}$ paramodulation step with 8. We get

10. $\neg Q[\alpha_{19} f5^{19\,19\,9}([], \alpha_{19}, \underline{\gamma})]$.

The empty clause is obtained if we resolve 10. with 3. using the appropriate unifier.

4.5 Conclusion

Properties of finite sets can be expressed in the logic of graded modalities. The usual inference calculi generate for all sets used in the proof at least as many constants (witnesses) as there are elements in each set. Even for moderate values a vast number of witnesses are generated which are processed by case distinctions in the proof.

In this chapter we presented an alternative method which avoids case distinctions, instead our method uses limited arithmetical reasoning. It arises in a series of transformation steps from the logic of graded modalities $\overline{K}$ into the new normal multi-modal logic $\overline{K}_E$, which reduces to first-order logic by the optimised functional translation. The point to note is that $\overline{K}_E$ does not reduce to first-order logic by the standard relational translation, because one of its axiom schemas is second-order. The optimised functional translation solves this irreducibility problem, as it exploits the richer structure of the functional models.

Our approach provides a viable alternative inference mechanism to the constraint (or tableaux) algorithms commonly used in the field of KL-ONE-based knowledge representation. The graded modal logic $\overline{K}$ is closely related to the description logic $\mathcal{ALC}$ with number restrictions, called $\mathcal{ALCN}$. In fact, there is an exact correspondence between terminological operators and modal operators (for details refer to Ohlbach et al. (1995, 1996) or van der Hoek and de Rijke (1995)).

The proposed approach of this chapter must be viewed as a first step toward efficient reasoning with finite sets. There are a number of open problems which need to be addressed.

One, a general completeness result for $\overline{K}_E$ will allow us to use the full expressivity of this system. The problematic schema $N7$ is like McKinsey's schema in that the technique of exchanging quantifiers bears a first-order property. It may just be that the theorems of Fine (1975) for uniform logics can be generalised appropriately for the completeness of $\overline{K}_E$. A uniform logic is an extension of KD and all axiom schemas are uniform formulae. McKinsey's schema and $N7$ are representatives of uniform formulae. For uniform logics Fine proves general completeness and finite model property theorems. As long as general completeness of $\overline{K}_E$ is not proved, we can guarantee completeness only for the original $\overline{K}$ formulae. This is what we wanted from the beginning, but a stronger result would be preferable.

Two, our first-order theory is represented by a set of axiom schemas which are understood to be conjunctions of all its instances with the numerical variables instantiated with concrete values. The implementation of the calculus will rely on theory resolution. The axiom schemas will be encoded as inference rules. Since the axiom schemas contain equations the realisation will not be easy, but it is certainly solvable.

Three, the original logic of graded modalities is decidable. Accordingly, we expect a resolution strategy for the translated formulae can be developed that is complete and terminates. This has yet to be done.

Four, the calculus is still limited in reasoning with arithmetical terms. It remains to be investigated whether and how this capability can be enhanced.

Five, we can apply our methods to KL-ONE-type reasoning but only for reasoning within the TBox, which corresponds directly to that in modal logic. We have not accounted for ABox reasoning about concrete instantiations of concepts/sets and roles/relations (ABox elements correspond to nominals in modal logic). The functional translation applied to ABox terms generates many equations. It is not immediate how these can be treated efficiently.

And six, although McKinsey's schema is not first-order definable by itself together with reflexivity and transitivity it is characterised by a first-order property, namely atomicity van Benthem (1984). It is open whether similarly, this is the case for the combination of $N7$ with of the other schemas of $\overline{K}_E$. We treated the schemas merely individually.

Chapter 5

Path logics and theory resolution

Our aim is to facilitate inference for modal logic by resolution on functional translations. This chapter defines a lattice of clausal logics, called *path logics*, into which the optimised functional translation maps normal modal logics, and it studies their closure under deduction with emphasis on theory unification.

Section 5.1 defines the lattice of path logics, of which *basic path logic* is the weakest. It is associated with the modal logics K, and $K_{(m)}$ as well as their serial extensions. Basic path logic extends to non-basic path logics by non-empty theories $\Upsilon\Pi_f(\Sigma)$ determined by the additional modal schemas in Σ. Path logics have two important properties: (i) their input clauses do not contain any Skolem terms other than Skolem constants, and (ii) variables in any clause have unique prefixes. Inference for basic path logics is facilitated by standard resolution with syntactic unification, which we briefly consider in Section 5.2. Sections 5.3 and 5.4 study syntactic unification and resolution, in particular, their effects on term lengths and the preservation of prefix stability. Inference for non-basic path logics with a theory is facilitated by theory resolution, introduced in Section 5.5. Then we address E-unification for T and 4 in Section 5.6, discussing decidability and mutation. In Section 5.7 we focus on the combination of T and 4 for which we present an improved E-unification calculus and prove preservation of prefix stability. The material on E-unification will appear in Schmidt (1998).

5.1 Path logics

The universal language of the lattice of path logics is a monadic first-order language with two principal sorts: the world sort W and the functional sort AF^* and its subsorts AF_i (for the different modalities). The vocabulary includes unary predicates, variables and constants of both sorts, one binary function $[\cdot,\cdot]$ for forming world terms, and functional operation symbols and the equality predicate $=$ as determined by the theory. The symbols $P, Q, \ldots$ denote unary predicates of which there are finitely many. There are possibly finitely many designated unary predicates de_i. Functional variables are denoted by Greek letters $\alpha, \beta, \ldots$ and functional constants by underlined Greek letters $\underline{\alpha}, \underline{\beta}, \ldots$. Functional variables and constants may be indexed by sorts. World variables are denoted by the Roman letters $x, y, z, \ldots$. There is a special world constant, denoted by the empty string $[]$ (for the initial world). $[\cdot,\cdot]$ is a binary left-associative operation and maps a world-function pair to a world term, called a *path*. Let u_i denote functional terms. Paths have the form

$$[[[[[]u_1]u_2]\dots]u_m] \quad \text{or} \quad [[[[xu_1]u_2]\dots]u_m],$$

or in shorthand notation

$$[u_1u_2\dots u_m] \quad \text{or} \quad [xu_1u_2\dots u_m].$$

The following notation and definitions will be used in subsequent chapters. The symbols $u, u_1, u_2, \dots$ and also $v, v_1, v_2, \dots$ are reserved for functional terms (when we consider basic path logic, they denote either functional variables or constants). We reserve the symbols u_0 and v_0 for the leading world constant $[]$ or world variable x in any path. The symbols $s, t, \dots$ are reserved for any world terms of the form $[]$ or $[u_1, u_2, \dots]$. Usually the term $[s\,t]$ is malformed since juxtaposition is assumed to be left-associative. When we write $[s\,t]$ we mean the term $[s\,u_1\dots u_i]$, where $t = [u_1\dots u_i]$. Given a world term $s = [u_0u_1u_2\dots u_m]$, define $s|_0 = u_0$ (that is, $s|_0 = []$ or x) and $s|_i = u_i$ for any $0 < i \leq m$.

The definition of prefixes in basic non-optimised path logic (as given in Section 2.5) extends naturally to path logics. Consider the term

$$s = [u_0u_1\dots u_iu_{i+1}\dots u_m].$$

(Often we omit u_0 and write just $s = [u_1\dots]$.) A *prefix in the world term s* is any subterm u_0 or $[u_1\dots u_i]$ (for $1 \leq i \leq m$). The *prefix of a functional term u_{i+1}* in a term s is the term $[u_0u_1\dots u_i]$. The world symbol u_0 has no prefix.

Similarly, we define suffixes. $[u_{i+1}\dots u_m]$ is a *suffix in the term s*. The *suffix of a functional term u_i* in the term s is $[u_{i+1}\dots u_m]$. u_m has no suffix. The *suffix of a subterm* $[u_0u_1\dots u_i]$ is $[u_{i+1}\dots u_m]$.

A set T of terms is said to be *prefix stable (for variables)* if any variable α occurring in T has exactly one prefix. A clause is said to be prefix stable (for variables) if the set of terms occurring (at argument position) in the clause has this property. The *prefix of a functional term u_i in a prefix stable clause* (or *set of terms*) is the unique prefix of u_i in any term of the clause (or set).

The following provides an alternative characterisation of prefix stability.[1]

Theorem 5.1.1 A set T of terms is prefix stable iff for any two terms $[u_0u_1\dots u_m]$ and $[v_0v_1\dots v_n]$ in T the conditions T1 and T2 hold for variables:

T1 If some variable u_i and some variable v_j are identical then $i = j$, and also

T2 the terms of each pair u_k and v_k preceding u_i and v_i, respectively, are also identical.

The proof is easy. T1 implies every variable u_i that occurs in a prefix stable term $s = [u_0\dots u_m]$ occurs exactly once in s. In other words, prefix stable terms are linear terms. This is an important fact that makes theory unification easier for many theories. T1 also implies every variable that occurs at position i in some term of the set T occurs at position i in every term, when it does occur in that term.

[1] This characterisation was inspired by the definition of tree-likeness found in Zamov (1989, p. 26).

We continue defining the syntactic constructs of path logics. Their logical connectives are $\wedge$, $\vee$ and $\neg$. By definition, a formula ψ is a *path formula* iff it is a conjunction of clauses that are prefix stable for variables.

The language of *basic path logic* for which the theory is empty is a restriction of this language. It has no world variables, $u_0 = []$, there are no compound functional terms and the equality symbol is not part of its language. *Path logics* are extensions of the basic path logic determined by a non-empty theory $\Upsilon\Pi_f(\Sigma)$. We will consider only first-order theories that are finite sets of clauses or equations, like those considered in Section 2.3.

This completes the syntactic definition of path logics. Their semantic models are given by Herbrand models. The general deduction calculus is theory resolution which will be discussed in the subsequent sections.

The class of path logics defined here should not be confused with the path logics of Auffray and Enjalbert (1992) or Fariñas del Cerro and Herzig (1995). There, path logics are the target logics of the non-optimised functional translation of quantified modal logics. We can transform any basic non-optimised path formula into a set of clauses of path logic: If ψ is any formula of non-optimised basic path logic then the clausal form of $\Upsilon(\psi)$ is well-formed in any path logic, provided the operation Υ moves all existential functional quantifiers inward over all universal quantifiers. The transformation in the inverse direction from path logic to non-optimised path logic is not always defined, because constants in clauses of path logic are not required to have unique prefixes. The restriction by prefix stability on the structure of paths is weaker than the restriction defined by the variable ordering in non-optimised path logics. However, the problem with prefix stability for constants is, it does not remain invariant under resolution.

Let c be a unary operator on first-order formulae such that $c(\psi)$ is a clausal form of ψ.

Theorem 5.1.2 Let φ be any modal formula and suppose it is not a schema. The set $S = c(\neg\Upsilon\Pi_f(\varphi))$ of clauses is well-formed in basic path logic (and thus in any path logic), provided the operation Υ moves all existential functional quantifiers inward over all universal quantifiers.

Proof. Routine, using Theorems 2.5.1 and 2.5.3. □

The transformation of any (non-optimised path) formula $\Pi_f(\varphi)$ into a set of path clauses can be realised by first forming the clausal form of the negation $\neg\Pi_f(\varphi)$ and then replacing all complex Skolem terms by constant symbols according to a mapping defined by

$$\cdot^{\Upsilon} : f(x_1, \ldots, x_n) \mapsto f.$$

Let c^{Υ} be a function from any set of first-order formulae to a set of path clauses that contain no complex Skolem terms defined by: $c^{\Upsilon}(\psi) = (c(\psi))^{\Upsilon}$. Evidently, for any modal formula φ,

$$S = c^{\Upsilon}(\neg\Pi_f(\varphi)) = c(\neg\Upsilon\Pi_f(\varphi)).$$

At worst the operation c produces a set of clauses with size that is exponentially larger than the size of the original formula. In Chapter 7 we will consider transformation routines that avoid the exponential increase in size and cause the output to grow by a linear factor. The functional translation Π_f causes a linear increase in size and the replacement by $\cdot^{\Upsilon}$ of Skolem terms by constants a linear decrease in size. Consequently, the entire transformation of a modal formula into path logic need have merely a linear overhead.

5.2 Resolution and condensing

The purpose of this section is to define briefly the basic inference rules of resolution procedures including condensing. A proper treatment can be found in Eisinger and Ohlbach (1993), Bachmair and Ganzinger (1997) or Leitsch (1997). Sample derivations were given in the Preview.

To begin with, some preliminary definitions: Atoms will be denoted by $A, B, A_1, A_2, \ldots$, literals by $L, L_1, L_2, \ldots$, clauses by $C, D, C_1, C_2 \ldots$, and sets of clauses by S, S'. The *complementary* (or *dual*) literal of a literal L is obtained by switching the sign and will be denoted by $L^\neg$. We regard any clause as a multi-set of literals. The complementary clause of $C = L_1 \vee \ldots \vee L_n$ is the clause $C^\neg = L_1^\neg \vee \ldots \vee L_n^\neg$. If we speak of logical relationships among clauses, like *logical equivalence* or *logical implication*, we regard clauses or sets of clauses as representing the corresponding closed formulae. A non-empty clause C is said to have a *most general unifier* σ when $C\sigma$, regarded as a set, is a singleton set. This means $L_1\sigma = \ldots = L_n\sigma$. (The most general unifier of an empty set is the identity mapping.) Two clauses are said to be *variants* of each other if they are equal modulo variable renaming. Two sets S and S' of clauses are *variants* of each other if every clause in S has a variant in S', and vice versa.

Inference according to the *resolution principle* employs three kinds of derivation steps: deduction, deletion and normalisation (of which deletion and normalisation can be regarded as redundancy elimination steps). Let $\triangleright$ be a binary relation on sets of sets of clauses that defines derivability in a resolution calculus. Deduction, deletion and normalisation rules have the following general form:

Deduction $\quad S \triangleright S \cup \{C\}$

provided C can be inferred from a subset of clauses in S.

Deletion $\quad S \cup \{C\} \triangleright S$

provided C is a redundant clause in $S \cup \{C\}$ with respect to S.

Normalisation $\quad S \cup \{C\} \triangleright S \cup \{N(C)\}$

provided $N(C)$ is a normal form of C with the property that C and $N(C)$ are logically equivalent.

Deduction based on the *resolution principle* (as defined by Robinson 1965) produces the resolvent $(C \vee C')\sigma$, from $C \vee D$ and $C' \vee D'$, provided the premises do not have common variables and $D \vee D'^\neg$ has a most general unifier σ. Resolution is usually implemented as a combination of binary resolution and factoring:

Binary resolution $\quad \dfrac{C \vee L \qquad C' \vee L'}{(C \vee C')\sigma}$

provided $C \vee L$ and $C' \vee L'$ do not share variables, and σ is a most general unifier of $L \vee L'^\neg$.

Factoring $\quad \dfrac{C \vee L_1 \vee \ldots \vee L_n}{(C \vee L_1)\sigma}$

provided σ is a most general unifier of $L_1 \vee \ldots \vee L_n$.

A clause $C\sigma$ is called a *factor* of $C \vee L$ if σ is a most general unifier of a subclause $D \vee L$ of at least two literals of $C \vee L$.

The *deletion strategy* demands deleting any redundant clauses whenever possible. Informally, a clause is redundant iff it is not needed for finding a proof (a contradiction). For example, a resolvent that is a variant of a clause derived earlier, is redundant and can be deleted. Also, a resolvent is redundant if it follows logically from and is more complex than a clause already in the input set. Of course, as testing logical equivalence is in general undecidable more efficient forms of redundancy elimination are employed in theorem provers. Standard theorem provers make use of subsumption deletion and additional strategies, like tautology deletion. A clause C *subsumes* another clause D iff there is a substitution σ such that $C\sigma$, regarded as a set, is a subset of D, also regarded as a set. When C subsumes D then C logically implies D, but not conversely. Subsumption deletion is defined by:

Subsumption deletion $\quad S \cup \{C, D\} \rhd S \cup \{C\}$
provided C subsumes D.

We will use a *normalising strategy* of replacing clauses by their condensations whenever possible. *Condensing* is due to Joyner Jr. (1976) and it is shown to be NP-hard by Gottlob and Fermüller (1993). A *condensation* of a clause C is a minimal subset of C which is also an instance of it. A clause is *condensed* if

$$\text{there is no substitution } \sigma \text{ such that } C\sigma \subsetneqq C,$$

that is, if it does not subsume a proper subclause of itself.[2] A condensation of a clause is logically equivalent to the clause. Because condensations are unique up to variable renaming we speak of *the* condensation of a clause C and denote it by $\textsc{cond}(C)$. The condensation of a set S of clauses is the set $\textsc{cond}(S) = \{\textsc{cond}(C) \mid C \in S\}$. Condensing is defined by the following normalising rule. It will be applied eagerly, that is, immediately after any resolution inference step.

Condensing $\quad S \cup \{C\} \rhd S \cup \{\textsc{cond}(C)\}$.

The condensation of a clause is a factor of that clause and consequently, condensing is available in any fair implementation of resolution with factoring and subsumption. (An implementation is regarded as being *fair* if no non-redundant inference is postponed forever.) Examples that demonstrate the COND operation can be found in the Preview.

Formally, a *resolution procedure* is a function $\mathcal{R}$ from finite sets of clauses to finite sets of clauses. More specifically, let S denote an input set (a finite set of finite clauses). $\mathcal{R}(S)$ is a finite subset of the union of S and the (possibly infinite) set of instances of resolvents of pairs of clauses in S. Let

$$\mathcal{R}^0(S) = S \quad \text{and}$$
$$\mathcal{R}^{n+1}(S) = \mathcal{R}(\mathcal{R}^n(S)), \quad \text{if } n > 0.$$

A resolution procedure $\mathcal{R}$ is (sound and) *complete*, provided the empty clause $\emptyset$ is in $\mathcal{R}^n(S)$ for some $n \geq 0$ iff S is unsatisfiable. For any complete resolution procedure $\mathcal{R}$, when for some $n \geq 0$, $\mathcal{R}^n(S)$ and $\mathcal{R}^{n+1}(S)$ are variants and the empty clause is not in $\mathcal{R}^n(S)$, then S is satisfiable. When $\mathcal{R}$ returns the set of resolvents defined by binary resolution and factoring we refer to $\mathcal{R}$ as *unrefined* or *unrestricted resolution*.

[2] Observe, minimality is with respect to size and the instance $C\sigma$ is regarded as a set.

By $\mathcal{R}_N$ we denote the combination of any complete resolution procedure $\mathcal{R}$ and any normalisation function N. Define $\mathcal{R}_N$ and $\mathcal{R}_N^n$ by:

$$\begin{aligned}
&\mathcal{R}_N(S) = S \cup N(\mathcal{R}(S) \setminus S),\\
&\mathcal{R}_N^0(S) = N(\mathcal{R}^0(S)) = N(S), \text{ and}\\
&\mathcal{R}_N^n(S) = \mathcal{R}_N(\mathcal{R}_N^{n-1}(S)) \quad \text{if } n > 0.
\end{aligned}$$

Note that $\mathcal{R}_N$ does not coincide with $\mathcal{R}_N^1$. For example, $\mathcal{R}_{\text{COND}}$ is the procedure $\mathcal{R}$ plus condensing which is applied eagerly. Joyner Jr. (1976) proves condensing is compatible with any resolution procedure that is complete via semantic trees. Condensing is an instance of the general redundancy criterion of Bachmair and Ganzinger (1994), and accordingly, condensing is also compatible with resolution procedures complete by the method of proof employing ordered minimal model extensions.

A note on our notation will be instructive. $\mathcal{R}$ denotes any complete resolution procedure. Later when we use theory resolution or ordering refinements this will be made explicit in the superscript. For example, $\mathcal{R}^E$ will denote any complete theory resolution procedure for the presentation E, and $\mathcal{R}^{\prec}$ will denote any complete resolution procedure restricted by the ordering $\prec$. The subscript will determine the normalisation operations used, for example COND or later we will define normalisation operations N_E which simplify terms according to given equational theories. $\mathcal{R}^{E,\prec}_{\text{COND}\,\circ\, N_E}$ denotes then an ordered E-resolution procedure with condensing and normalisation by N_E. Moreover, $\mathcal{R}^{E,\prec}_{\text{COND}\,\circ\, N_E}$ and $(\mathcal{R}^{E,\prec}_{\text{COND}\,\circ\, N_E})^n$ are defined like $\mathcal{R}_N$ and $\mathcal{R}_N^n$ with $\mathcal{R}^{E,\prec}$ instantiated for $\mathcal{R}$ and COND $\circ\, N_E$ for N.

5.3 Basic path unification and term depths

In this section we consider the effect of the language restrictions of basic path logic on substitutions, unifiers and path lengths (or term depths).

Any non-empty substitution σ defined over sets of terms in the basic path logic consists of bindings that have one of two forms

$$\alpha \mapsto \beta \quad \text{or} \quad \alpha \mapsto \underline{\gamma}, \tag{5.1}$$

where α and β are variables and $\underline{\gamma}$ is a constant, because the basic path language has no compound functional terms. A substitution is said to be *admissible* for basic path logic iff its bindings have the form (5.1).

The general transformation rules of syntactic tree based unification (for example, from Jouannaud and Kirchner 1991) adapt to those of Figure 5.1 for basic path logic. P denotes a problem set of pairs $s =^? t$ of world terms or $u =^? v$ of functional terms. The symbol $\rightsquigarrow$ denotes the derivability relation in a unification calculus. $\perp$ indicates failure of the unification problem. All other symbols have the usual interpretation, and s and t may be empty paths. Evidently, any most general unifier of a unification problem over paths without world variables obtained by the rules of Figure 5.1 is an admissible substitution. As no world variables occur in basic path clauses, the occurs check rule is superfluous. Soundness and completeness is immediate by soundness and completeness of the general rules for syntactic unification.

$$
\begin{array}{llcl}
\textbf{Delete} & P \cup \{s =^? s\} & \rightsquigarrow & P \\
 & P \cup \{u =^? u\} & \rightsquigarrow & P \\
\textbf{Decompose} & P \cup \{[su] =^? [tv]\} & \rightsquigarrow & P \cup \{u =^? v,\ \ s =^? t\} \\
\textbf{Conflict} & P \cup \{[su] =^? []\} & \rightsquigarrow & \bot \\
 & P \cup \{\underline{\alpha} =^? \underline{\beta}\} & \rightsquigarrow & \bot \\
 & \text{provided } \underline{\alpha} \neq \underline{\beta} & & \\
\textbf{Coalesce} & P \cup \{\alpha =^? \beta\} & \rightsquigarrow & P\{\alpha \mapsto \beta\} \cup \{\alpha =^? \beta\} \\
 & \multicolumn{3}{l}{\text{provided } \alpha \neq \beta \text{ are variables both occurring in } P.} \\
\textbf{Eliminate} & P \cup \{\alpha =^? \underline{\beta}\} & \rightsquigarrow & P\{\alpha \mapsto \underline{\beta}\} \cup \{\alpha =^? \underline{\beta}\} \\
 & \multicolumn{3}{l}{\text{provided } \alpha \text{ is a variable occurring in } P \text{ and } \underline{\beta} \text{ is constant.}}
\end{array}
$$

Figure 5.1: Syntactic unification rules for basic path logic

Theorem 5.3.1 Let P be a set of pairs $s =^? t$ of paths. Repeatedly applying the above rules to P yields $\bot$ iff P is not unifiable, and a tree solved form $\{\alpha_1 =^? u_1, \ldots, \alpha_n =^? u_n\}$ iff $\{\alpha_1 \mapsto u_1, \ldots, \alpha_n \mapsto u_n\}$ is a most general unifier of P.

Recall, in general, a *tree solved form* is any set of equations $\{\alpha_1 =^? u_1, \ldots, \alpha_n =^? u_n\}$ possibly with variables $\beta_1, \ldots, \beta_m$ introduced during unification that do not occur in the original problem set P, such that (i) any α_i is given only one value (formally, $\alpha_i \neq \alpha_j$ for any $1 \leq i < j \leq n$), (ii) the value for any α_i is a finite term (α_i does not occur in any u_j), (iii) no α_i coincides with a new variable β_j, and (iv) the new variables are useful and contribute to the values of other variables (any β_i occurs in some u_j).

It is important that we keep in mind any term $[u_0 u_1 \ldots u_m]$ (with or without u_0) is an abbreviation for a nested term $[[[[[]u_1]u_2]\ldots]u_m]$. This determines how paths are decomposed, namely from right to left. The following is one way of deriving a most general unifier for the terms $[\underline{\alpha}\beta\underline{\gamma}]$ and $[\underline{\alpha}\delta\epsilon]$ (from $c(\neg\Upsilon\Pi_f(\rho_1))$ of the Preview).

$$
\begin{array}{l}
\{[\underline{\alpha}\beta\underline{\gamma}] =^? [\underline{\alpha}\delta\epsilon]\} \overset{\text{Dec.}}{\rightsquigarrow} \{\underline{\gamma} =^? \epsilon,\ [\underline{\alpha}\beta] =^? [\underline{\alpha}\delta]\} \\
\overset{\text{Dec.}}{\rightsquigarrow} \{\underline{\gamma} =^? \epsilon,\ \beta =^? \delta,\ [\underline{\alpha}] =^? [\underline{\alpha}]\} \overset{\text{Del.}}{\rightsquigarrow} \{\underline{\gamma} =^? \epsilon,\ \beta =^? \delta\}.
\end{array}
$$

We did not use the rules Coalesce and Eliminate. In fact:

Theorem 5.3.2 Let P be a singleton set $\{s =^? t\}$ with s and t terms for which T1 for variables holds. Then the rules Coalesce and Eliminate are redundant.

Proof. These rules are not applicable to s and t when s and t are both constants or when one is a variable and the other is a constant.

Assume $s = [u_0 u_1 \ldots u_m]$ and $t = [v_0 v_1 \ldots v_n]$ for both m and n non-zero values. The only applicable rule is Decompose. $\{s =^? t\}$ becomes

$$(*) \qquad \{[u_0 u_1 \ldots u_{m-1}] =^? [v_0 v_1 \ldots v_{n-1}],\ \ u_m =^? v_n\}.$$

Suppose u_m is a variable and $u_m \neq v_n$. Delete is not applicable. Since T1 holds for s (and t) no variable occurs more than once in s (and t). Then $u_m \neq u_i$ for all $i < m$.

(i) If s and t are variable disjoint, then also $u_m \neq v_j$ for all $j < n$. This means the condition of both Coalesce and Eliminate that u_m is a variable in $(*)$ without $u_m =^? v_n$ is not true.

(ii) If s and t have common variables then it can happen that v_m coincides with u_m if $m < n$. In this case s and t do not have equal length and are not unifiable because a conflict of the form $s =^? []$ will occur with s non-empty, regardless of whether or not we apply the rules Coalesce and Eliminate. (This is another argument why s and t are not unifiable if they have different lengths.)

Splitting any P in the search tree into two sets $P_1 = \{s =^? t\}$ and $P_2 = \bigcup\{u =^? v\}$, it is not difficult to prove the result by an inductive argument on the lengths of the paths in P_1. □

Now, we briefly consider term depths. The functional *depth* of a term s, written $\tau(s)$, is defined as usual. Consequently basic paths of the form $[u_0 u_1 \ldots u_m]$ have depth $m + 1$. We find it convenient to speak of the term *length* of a basic path $[u_0 u_1 \ldots u_m]$ with $u_0 = []$, which we set to m.

Since any admissible substitution does variable renaming or instantiation with a constant, it is immediate that no admissible substitution or unifier changes the lengths (or depths) of paths. Also, no admissible substitution or unifier increases the number of variables in terms. On the contrary, $s\sigma$ may have fewer variables that s. Since substitutions preserve path lengths, so does factoring. It is easy to prove that resolution does not increase path lengths. Consequently:

Theorem 5.3.3 The depth of all literals (or length of all terms) in $\mathcal{R}^n(S)$ has an upper bound, namely, the maximum of the depths of all literals in S:

$$\tau(\mathcal{R}^n(S)) \leq \tau(S).$$

This establishes that for basic path logic the level of functional nesting is bounded and the bound is the maximum level of nesting in the input clauses. We have verified the easier part, namely (A) of the Preview, required for the proof of decidability for basic path logic.

The situation is pleasantly simple for the logic *S5*. In *S5* any sequence of modal operators can be replaced by the first one in the sequence, and *S5* corresponds to the fragment of monadic first-order logic in one variable (via the relational translation). This is reflected in the corresponding path logic by the fact that any singleton unification problem $[u_1 \ldots u_m] =^? [v_1 \ldots v_n]$ can be seen to reduce to the unification problem of $[u_1] =^? [v_1]$. Such problems can be solved modulo (a subset of) the rules of Figure 5.1.

5.4 Prefix stability and resolution

This section verifies that basic path logic is closed under the operations of resolution, factoring, subsumption deletion and condensing, which amounts to showing the operations preserve prefix stability for variables. The theorems and proofs are in line with the preservation results outlined in Ohlbach (1988a, 1991) and Zamov (1989) for tree-likeness.

The variables as well as the constants in any clause of $c(\neg\Upsilon\Pi_f(\varphi))$ have unique prefixes, which means that T1 and T2 of Theorem 5.1.1 are true for variables as well as for constants. However, forming resolvents does not preserve the property T2 for constants, which is the reason why terms in path logic are merely required to satisfy prefix stability for variables. For example, the resolvent of

$$\neg P[\alpha\beta] \vee Q[\alpha\beta\underline{\gamma}]$$
$$P[\alpha'\underline{\epsilon}] \vee \neg P[\alpha'\delta\underline{\gamma}]$$

which both satisfy T1 and T2 for variables as well as for constants, is

$$Q[\alpha\underline{\epsilon}\underline{\gamma}] \vee \neg P[\alpha\delta\underline{\gamma}].$$

The constant $\underline{\gamma}$ has two different prefixes, namely $[\alpha\underline{\epsilon}]$ and $[\alpha\delta]$.

Arbitrary (admissible) substitution does not preserve T1 (neither for variables nor for constants) or T2. Take the clause $P[\alpha\beta\gamma]$ and apply the substitution $\{\gamma \mapsto \alpha\}$. The result is $P[\alpha\beta\alpha]$ and does not satisfy T1. Evidently, variable renaming does no harm: Any variant of a clause (or set of terms) that satisfies T1 and T2 for variables (and/or for constants) does, too.

In the following we prove that the basic path logic is closed under the application of a unifier, forming subsets, factoring, condensing, forming disjoint unions, and most important, resolution. We assume T is a set of terms in the vocabulary of basic path logic. By definition, two paths s and t (of equal length) are *k-equal* if s and t are equal except possibly at position k, that is, for every position $i \neq k$, $s|_i = t|_i$.

Theorem 5.4.1 Let $[u_0 u_1 \ldots u_m]$ and $[v_0 v_1 \ldots v_n]$ be two terms in T such that for some $k > 0$,

$$u_0 = v_0, u_1 = v_1, \ldots, u_{k-1} = v_{k-1} \quad \text{and} \quad u_k \neq v_k \tag{5.2}$$

and u_k is a variable. Let σ be the substitution $\{u_k \mapsto v_k\}$. Then $T\sigma$ satisfies T1 and T2, provided T does.

Proof. We consider two arbitrary terms in $T\sigma$. They are of the form $s\sigma$ and $t\sigma$ with s and t some terms in T. For s and t conditions T1 and T2 hold. We want to show they hold for the terms $s\sigma$ and $t\sigma$, too.

Lemma 5.4.2 The terms $s\sigma$ and s are k-equal and differ only when $s|_k = u_k$.

Proof. σ affects only the variable u_k and in any term of T, u_k occurs only at position k else condition T1 is violated. Hence, if u_k occurs in s then $s|_k = u_k$ and for any $l \neq k$, $s|_l \neq u_k$. In this case $s\sigma|_k = v_k \neq s|_k$. □

We continue the proof of Theorem 5.4.1. The lemma is true for $t\sigma$ and t, as well. If neither s nor t contain the variable u_k then the substitution σ does not affect s and t. Then $s\sigma = s$ and $t\sigma = t$. In this case $s\sigma$ and $t\sigma$ trivially satisfy T1 and T2 (since s and t do).

Therefore, we assume without loss of generality that $s|_k = u_k$. Then $s|_k\sigma = v_k$. Distinguish two cases:

(i) $t|_k \neq u_k$ and $t|_k \neq v_k$. σ leaves t unchanged so that $t\sigma = t$. Suppose $s\sigma|_i = t\sigma|_j$ is a variable. Then $s\sigma|_i = t\sigma|_j = t|_j$. Also, $j \neq k$ and $i \neq k$, since otherwise $t|_j = v_k$ which contradicts our assumption. This implies $s|_i = s\sigma|_i = t\sigma|_j = t|_j$. By T1 which holds for s and t we get $i = j$. By T2 for any $l < i = j$ we have $s|_l = t|_l$. Hence $i = j < k$ since otherwise, if $i = j = k$ then $s|_i = u_k \neq t|_i$ which is a contradiction, or if $i = j > k$ then since $s|_k \neq t|_k$ by assumption, s and t contradict T2. Consequently by the Lemma $s\sigma|_l = s|_l = t|_l = t\sigma|_l$. Therefore, conditions T1 and T2 are true for case (i).

(ii) Now we consider the case that $t|_k = u_k$ or $t|_k = v_k$. Then $t\sigma|_k = v_k = s\sigma|_k$. Suppose $s\sigma|_i = t\sigma|_j$ is a variable.

a. If $i = k$ then $s\sigma|_i = v_k = t\sigma|_j$. Then, either $t|_j = u_k$ or $t|_j = v_k$. In either case, it follows that $j = k$ and hence $i = j$.

b. If $j = k$ then by a similar argument $i = j$.

c. If $i \neq k$ and $j \neq k$ then the Lemma implies $s\sigma|_i = s|_i$ and $t\sigma|_j = t|_j$. Since $s\sigma|_i = t\sigma|_j$ we have $s|_i = t|_j$ and it follows by T1 that $i = j$.

Therefore, $s\sigma$ and $t\sigma$ satisfy T1.

Let $l < i = j$ be arbitrary. By T2 we have $s|_l = t|_l$.

a. Consider the case that $l \neq k$. By the Lemma $s\sigma|_l = s|_l$ and $t\sigma|_l = t|_l$. Since $s|_l$ and $t|_l$ coincide, we conclude $s\sigma|_l = t\sigma|_l$. (Note that if $t|_k = v_k$ then $s|_k = u_k \neq t|_k$ and consequently $i = j < k$.)

b. For $l = k$: $s\sigma|_l = v_k = t\sigma|_l$ by assumption.

This completes the proof. □

Theorem 5.4.3 Let σ be an idempotent unifier of two terms s and t in T. If T satisfies properties T1 and T2 then so does $T\sigma$.

Proof. Because σ is an idempotent unifier it coincides with a composition $\sigma_1 \ldots \sigma_l$ of bindings with $\sigma = \bigcup_i \sigma_i$. The order of the application of the bindings in σ is arbitrary. We choose exactly those bindings $\sigma_1, \ldots, \sigma_n$ that do affect s and t and arrange them so that the order of application of the σ_i is determined by the position of the variable in the domain of σ_i in the terms s and t. We let $\sigma_1 \ldots \sigma_n$ be such that for every pair of bindings σ_i and σ_j with $1 \leq i < j \leq n$, the variable in the domain of σ_i is associated with a position strictly less than the position associated with the variable in the domain of σ_j. We have $s\sigma = s\sigma_1 \ldots \sigma_n = t\sigma = t\sigma_1 \ldots \sigma_n$. Now, s and t have equal length. Suppose $s = [u_0 u_1 \ldots u_m]$ and $t = [v_0 v_1 \ldots v_m]$. s and t satisfy the condition (5.2) of the previous theorem with k being the first position at which s and t differ. In other words, k is the position associated with the variable in the domain of σ_1. k is greater than 0. Consequently by Theorem 5.4.1, $T\sigma_1$ satisfies T1 and T2. Now use an inductive argument to see that $T\sigma_1 \ldots \sigma_n = T\sigma$ satisfies T1 and T2. □

This generalises to:

Corollary 5.4.4 Let σ be the most general unifier of a non-empty subset T' of T. If T satisfies properties T1 and T2 then so does $T\sigma$.

Proof. If T' is a singleton set then σ is empty and hence $T\sigma = T$. Suppose T' has two or more elements. Choose any two different terms s and t in T'. σ is an idempotent unifier of s and t. Therefore, by the previous result, $T\sigma$ satisfies T1 and T2. □

Thus, basic path logic is closed under factoring. Trivially it is also closed under subsumption deletion. The following is easy to see, which proves basic path logic is closed under condensing and $\textsc{cond}(S)$ is well-formed in basic path logic.

Theorem 5.4.5 If T satisfies properties T1 and T2 then so does

(i) any subset T' of T, and consequently also

(ii) any condensation T' of T.

Theorem 5.4.6 Let T and T' be two sets of terms in the vocabulary of basic path logic that have no variables in common. If T and T' satisfy T1 and T2, then their union $T \cup T'$ satisfies T1 and T2, too.

Proof. Let s and t be two terms in $T \cup T'$. Suppose $s|_i = t|_j$ and $s|_i$ is a variable. Then both s and t are either both in T or they are both in T'. Hence T1 and T2 are true, since they are for T and for T'. □

The preservation result for resolution is a direct consequence:

Theorem 5.4.7 The binary resolvent of two clauses satisfying T1 and T2 that have no common variables also satisfies T1 and T2.

Proof. Let C_1 and C_2 be two clauses that have no common variables satisfying T1 and T2. Suppose C is a binary resolvent of C_1 and C_2 with L_1 and L_2 being the literals resolved upon. Let σ be the most general unifier of L_1 and $L_2^{\neg}$. Let T_1, T_2 and T be the sets of terms that occur in C_1, C_2 and C, respectively. T_1 and T_2 satisfy T1 and T2 and by the previous result their union $T_1 \cup T_2$ does too. σ is the most general unifier of a subset of $T_1 \cup T_2$. Hence, by Corollary 5.4.4, $(T_1 \cup T_2)\sigma = T$ satisfies T1 and T2. Consequently the resolvent C satisfies T1 and T2. □

Corollary 5.4.8 Let S be a set of clauses in basic path logic. Then:

(i) Both $\mathcal{R}(S)$ and $\mathcal{R}_{\textsc{cond}}(S)$ are well-formed in basic path logic.

(ii) For any n, both $\mathcal{R}^n(S)$ and $\mathcal{R}^n_{\textsc{cond}}(S)$ are well-formed in basic path logic.

Using Theorem 5.1.2, the fundamental preservation result follows:

Theorem 5.4.9 Let φ be any modal formula and $S = c(\neg\Upsilon\Pi_f(\varphi))$. Then, for any n, $\mathcal{R}^n(S)$ and $\mathcal{R}^n_{\textsc{cond}}(S)$ are well-formed in basic path logic.

5.5 Theory resolution

The reader is assumed to have some familiarity with theory resolution introduced by Stickel (1985) and E-unification introduced by Plotkin (1972). A survey paper on resolution in which theory resolution is treated as well, is Eisinger and Ohlbach (1993). Good survey papers on unification including theory unification are Baader and Siekmann (1993), Gallier and Snyder (1992) and Jouannaud and Kirchner (1991), and the paper by Dershowitz and Jouannaud (1990) also has a section on E-unification.

For later reference we first recall some definitions and known general facts. Let T be a theory, given by a set of clauses. A T-model is any model satisfying T. A formula ψ is *T-satisfiable* (respectively *T-unsatisfiable*) iff it is true in some (resp. no) T-model. A substitution σ is a *T-unifier* of a set or multi-set of terms (or literals) iff for any pair of terms s and t (or literals L and L') in the set $s\sigma$ and $t\sigma$ (or $L\sigma$ and $L'\sigma$) are equivalent under T. When T is an equational theory we use the notation E, E-model, etcetera.

In this thesis we focus on *total theory resolution*, and as a special case resolution under an equational theory E. The rules of our calculus are total binary theory resolution and standard factoring defined by:

Binary theory resolution $\dfrac{C \vee L \qquad C' \vee L'}{(C \vee C')\sigma}$

provided $C \vee L$ and $C' \vee L'$ do not share variables, and σ is a most general T-unifier of $L \vee L'^{\neg}$.

Factoring $\dfrac{C \vee L_1 \vee \ldots \vee L_n}{(C \vee L_1)\sigma}$

provided σ is the most general (syntactic) unifier of $L_1 \vee \ldots \vee L_n$.

Most general T-unifiers are not necessarily unique. Any set of literals may have one, finitely many, infinitely many most general T-unifiers, or none at all. The computation of most general T-unifiers will be implemented in separate unification algorithms. For completeness the procedure must generate a complete (not necessarily minimal) set of most general T-unifiers.[3] Unfortunately, there is no general recipe, and for every theory a complete unification procedure has to be developed anew. As to ensure completeness of the resolution procedure an important requirement of theory resolution is that the unification algorithms are semi-decision procedures.

We will focus on resolution under equational theories and present E-unification algorithms that are decision procedures. The theories we will consider have the pleasant property that they have normal forms, which is to say, two terms s and t are E-equivalent iff their normal forms coincide. Plotkin (1972) showed that E-resolution may be restricted to one representative of any E-equivalence class, which we realise by a normalisation rule:

Normalisation under E $\quad S \cup \{C\} \triangleright S \cup \{N_E(C)\}$.

N_E is a recursive function from terms to their normal forms with respect to E. By this we mean N_E is a simplification function on the terms in the language determined by E with the property that $t =_E N_E(t)$, and if $s =_E t$ then $N_E(s) = N_E(t)$. The application of N_E to literals and clauses is given by:

$$\begin{aligned} N_E(\pm Ps) &= \pm P(N_E(s)), \\ N_E(C \vee L) &= N_E(C) \vee N_E(L) \quad \text{and} \\ N_E(S) &= \{N_E(C) \mid C \in S\}. \end{aligned}$$

Completeness of unrestricted resolution under an equational theory with and without normalisation by N_E is proved in Plotkin (1972). Given a complete E-resolution procedure

[3] We do not treat infinitary theories. For infinitary theories the procedure must enumerate a complete set of most general T-unifiers.

$\mathcal{R}^E$ we define $\mathcal{R}^E_{N_E}$ in analogy to $\mathcal{R}_N$. In Chapter 7 we will prove completeness for ordered E-resolution from which it also follows that unordered resolution by $\mathcal{R}^E_{N_E}$ is complete. Observe that N_E simplifies terms and differs from normalisation by condensing which simplifies clauses (by eliminating literals). We will use normalisation to eliminate functional subterms, like $\underline{e}$ and $u \circ v$.

5.6 Unification for T and 4

In this section we focus on equational theories, which we can, by assuming seriality. By E we will denote any finite presentation $\Upsilon\Pi_f(\Sigma)$ with D in Σ. In particular, the presentations we consider include one of the equations

$$\begin{aligned} &\textit{right identity}: && [x\underline{e}] = x \\ &\textit{associativity}: && [x(\alpha \circ \beta)] = [x\alpha\beta], \end{aligned}$$

or combinations thereof. It goes without saying, these presentations are consistent, as required. In general, E-unification is *finitary* iff for any solvable unification problem a minimal complete set of unifiers exists and the set is finite. We will refer to unifiers under right identity, associativity or both, as *T-unifiers*, *4-unifiers* or *S4-unifiers*, respectively.

Unification of paths under the identity law is finitary and decidable. This can be easily seen by considering a unification problem in n variables and forming 2^n syntactic unification problems by replacing some of the variables by $\underline{e}$. Each of the problems is decidable by syntactic unification in linear time. Therefore, unification under right identity is in NP. By a result of Arnborg and Tidén (1985) for standard right identity it is at worst NP-complete.

Unification under our form of associativity is related to unification under standard associativity. In (1972) Plotkin showed unification in free semi-groups is infinitary. He also gives a unification algorithm that is sound and complete, but it is not guaranteed to terminate. There are decision procedures for general A-unification by Makanin (1977) and Jaffar (1990), for example, but these are far too complex for our purposes. Fortunately, though Plotkin's algorithm is non-terminating in the general case, it decides unification problems of one linear equation, or one equation in which no variable occurs more than twice (Schulz 1992). This implies, unification of paths under the form of associativity we consider is also finitary (via an appropriate isomorphism). (By exploiting the correspondence to regular expressions and using methods from automata theory, we expect that unifiability under identity and/or associativity can be decided in polynomial time.)

Unification algorithms are described in Ohlbach (1988a, 1991), Fariñas del Cerro and Herzig (1995) and Auffray and Enjalbert (1992) for the non-optimised translation of quantified modal logics and in Zamov (1989) for the non-optimised translation of propositional *S4*. The first three algorithms are not complete for a minor omission. Problems of the form $\{s =^? s * \underline{e}\}$, $\{s =^? s\,;\textsc{id}\}$ and $\{s\,!\,\alpha\,!\,f(s\,!\,\alpha) =^? s\,!\,f(s)\}$ (using in essence the notation of the respective authors) are not treated properly, which can be rectified by adding a rule for deleting the identity constant.

Ohlbach's rules improve the respective instances of lazy paramodulation. For paths the general lazy paramodulation rule of Gallier and Snyder (1992) simplifies to:

Lazy paramod. $P \cup \{[ss'] =^? t\} \quad \rightsquigarrow \quad P \cup \{s =^? l, [rs'] =^? t\}$
provided s is not a variable and $l = r$ is (a variant of) an equation in E or $E^{\smile}$.

Delete	$P \cup \{s =^? s\}$	$\leadsto$	P
Decompose	$P \cup \{s * s' =^? t * t'\}$	$\leadsto$	$P \cup \{s =^? t,\ \ s' =^? t'\}$
Check	$P \cup \{\alpha =^? s\}$	$\leadsto$	$\perp$
	when s is not a variable and α occurs in s		
Coalesce	$P \cup \{\alpha =^? \beta\}$	$\leadsto$	$P\{\alpha \mapsto \beta\} \cup \{\alpha =^? \beta\}$
	when $\alpha \neq \beta$ are variables both occurring in P		
Eliminate	$P \cup \{\alpha =^? s\}$	$\leadsto$	$P\{\alpha \mapsto s\} \cup \{\alpha =^? \underline{\beta}\}$
	when α is a variable occurring in P and s is not a variable		
Identity	$P \cup \{s * \alpha * s' =^? t\}$	$\leadsto$	$P \cup \{\alpha =^? \underline{e},\ \ s * s' =^? t\}$
Path-separat.	$P \cup \{\alpha * s =^? t * t'\}$	$\leadsto$	$P \cup \{\alpha =^? t,\ \ s =^? t'\}$
Splitting	$P \cup \{\alpha * s'' * s =^? t * t'' * \beta * t'\}$		
	$\leadsto P \cup \{\alpha =^? t * t'' * \beta_1,\ \ \beta =^? \beta_1 * \beta_2,\ \ s'' * s =^? \beta_2 * t'\}$		
	when s'' and t'' are non-empty and β_1, β_2 are new variables.		

Figure 5.2: Ohlbach's unification rules for T and 4

Both s and s' are either empty or have the form $[s''u]$. An additional condition results improves the computational behaviour (Jouannaud and Kirchner 1991, Baader and Siekmann 1993). Namely, when l is not a variable then s and l have the same root symbol and Decompose is applied to $s =^? l$ directly after an application of lazy paramodulation. The appropriate combination of rules for T and 4 together with the non-failing rules for syntactic unification (Delete, Decompose, Coalesce and Eliminate) provide a complete system for computing a complete set of unifiers for a unification problem in the corresponding E.

Figure 5.2 presents a reformulation of the algorithms from Ohlbach (1991) relevant for propositional *S4* in terms of rules. Ohlbach's language is a variation from ours. Terms are lists built from variables and constants of the sort AF with an associative operation $*$, making the additional operation $\circ$ superfluous. (The correspondence to world terms of basic path logic is given by $h([[]u]) = u$ and $h([su]) = h(s) * u$ when $s \neq []$.) The symbols s, s', t and t' in the figure denote (possibly empty) lists. It turns out, the deletion rule for identity is not required for solving singleton problems, so that under this condition the system is complete.

By way of an example we will demonstrate the system can be improved. Figure 5.3 sketches a derivation of *S4*-unifiers for $\{\alpha * \underline{\beta} * \underline{\gamma} * \delta =^? \underline{\alpha} * \beta * \underline{\gamma}\}$. Terms are decomposed from left to right. Failure branches are those that do not produce solved forms, that is, sets of the form $\{\alpha_1 =^? u_1, \ldots, \alpha_n =^? u_n\}$ with each α_i occurring exactly once in the set. The successful branches in the derivation tree are those marked with numbers, whose solved forms yield the following unifiers:

1. $\{\alpha \mapsto \underline{\alpha},\ \ \beta \mapsto \underline{\beta},\ \ \delta \mapsto []\}$
2. $\{\alpha \mapsto \underline{\alpha},\ \ \beta \mapsto [\underline{\beta}\underline{\gamma}],\ \ \delta \mapsto \underline{\gamma}\}$
3. $\{\alpha \mapsto \underline{\alpha},\ \ \beta \mapsto \underline{\beta},\ \ \delta \mapsto []\}$ (which coincides with 1.)
4. $\{\alpha \mapsto \underline{\alpha},\ \ \beta \mapsto [\underline{\beta}\underline{\gamma}\delta_1],\ \ \delta \mapsto [\delta_1\underline{\gamma}]\}$

$\vdots$

$$
\begin{array}{l}
[\alpha\underline{\beta}\underline{\gamma}\delta] =^? [\underline{\alpha}\beta\underline{\gamma}] \\
\overset{\text{Dec.}}{\leadsto} \quad \alpha =^? \underline{\alpha},\ [\underline{\beta}\underline{\gamma}\delta] =^? [\beta\underline{\gamma}] \\
\qquad\qquad \left.\begin{array}{l}
\overset{\text{Dec}}{\leadsto} \quad \beta =^? \underline{\beta},\ [\underline{\gamma}\delta] =^? [\underline{\gamma}] \\
\qquad\qquad \overset{\text{Id}}{\leadsto} \quad \delta =^? [] \quad \leadsto \quad 1. \\
\qquad\qquad \overset{\text{Dec}}{\leadsto} \quad \delta =^? [] \quad \text{redundant} \\
\overset{\text{Id}}{\leadsto} \quad \beta =^? [],\ [\underline{\beta}\underline{\gamma}\delta] =^? \underline{\gamma} \quad \leadsto \quad \ldots \quad \text{not solved} \\
\overset{\text{Sep}}{\leadsto} \quad \beta =^? [\underline{\beta}\underline{\gamma}],\ \delta =^? \underline{\gamma} \quad \leadsto \quad 2. \\
\overset{\text{Sep}}{\leadsto} \quad \beta =^? \underline{\beta},\ [\underline{\gamma}\delta] =^? \underline{\gamma} \quad \leadsto \quad \ldots \quad \text{not solved} \\
\overset{\text{Sep}}{\leadsto} \quad \beta =^? [],\ [\underline{\beta}\underline{\gamma}\delta] =^? \underline{\gamma} \quad \text{redundant} \\
\overset{\text{Id}}{\leadsto} \quad \delta =^? [],\ [\underline{\beta}\underline{\gamma}] =^? [\beta\underline{\gamma}] \\
\qquad\qquad \overset{\text{Dec,Del}}{\leadsto} \quad \beta =^? \underline{\beta} \quad \leadsto \quad 3. \\
\qquad\qquad \overset{\text{Id}}{\leadsto} \quad \beta =^? [],\ [\underline{\beta}\underline{\gamma}] =^? [\underline{\gamma}] \quad \text{not solved} \\
\qquad\qquad \overset{\text{Sep}}{\leadsto} \quad \beta =^? [],\ [\underline{\beta}\underline{\gamma}] =^? [\underline{\gamma}] \quad \text{redundant} \\
\overset{\text{Split}}{\leadsto} \quad \beta =^? [\underline{\beta}\underline{\gamma}\delta_1],\ \delta =^? [\delta_1\delta_2],\ \delta_2 =^? \underline{\gamma} \quad \leadsto \quad 4.
\end{array}\right\} (*) \\
\overset{\text{Split}}{\leadsto} \quad \alpha =^? [\underline{\alpha}\beta_1],\ \beta =^? [\beta_1\beta_2],\ [\underline{\beta}\underline{\gamma}\delta] =^? [\beta_2\underline{\gamma}] \\
\qquad\qquad\qquad\qquad \vdots \ \text{like } (*) \\
\overset{\text{Id}}{\leadsto} \quad \alpha =^? [],\ [\underline{\beta}\underline{\gamma}\delta] =^? [\underline{\alpha}\beta\underline{\gamma}] \quad \leadsto \quad \ldots \\
\overset{\text{Id}}{\leadsto} \quad \delta =^? [],\ [\alpha\underline{\beta}\underline{\gamma}] =^? [\underline{\alpha}\beta\underline{\gamma}] \quad \leadsto \quad \ldots \\
\overset{\text{Id}}{\leadsto} \quad \beta =^? [],\ [\alpha\underline{\beta}\underline{\gamma}\delta] =^? [\underline{\alpha}\underline{\gamma}] \quad \leadsto \quad \ldots
\end{array}
$$

Figure 5.3: A sample derivation of *S4*-unifiers according to Ohlbach's algorithm

We will give a set of rules applying paramodulation only at the top symbols of the terms of an equation $s =^? t$ bearing a more efficient unification algorithm. These restricted forms of paramodulation rules are known as *mutation rules* and are sound and complete only for very particular E. For instance, they may be applied where E (is a finite resolvent set of equations and) defines a syntactic theory. In a syntactic theory a proof of satisfiability of two terms can always be performed with at most one step at the top of s or t. Two results from the literature are of interest.

(i) Kirchner and Klay (1990) prove mutation rules are complete for syntactic collapse-free theories.

(ii) Comon et al. (1994) consider mutation with (and without) collapsing equations for shallow theories and prove any shallow theory is syntactic.

A *collapsing equation* has the form $x = t$ with x a variable of t and $x \neq t$. A theory E is *collapse-free* if no presentation of E contains a collapsing equation. A *shallow* theory has a finite presentation of shallow equations, defined to be equations $s = t$ with all variables of s or t occurring at depth one. This is relevant for the identity law which is collapsing and shallow. The result of Kirchner and Klay is relevant for our associativity law which can be shown to be syntactic by an analogous argument as in Klay (1991) for ordinary associativity. (Among the theory equations we derived in an earlier chapter the equation $[x\alpha] = [x\beta]$ associated with $\mathit{Funct} = \Diamond p \to \Box p$ is also shallow.)

There are a number of negative results concerning termination for algorithms with mutation rules. Klay (1991) showed collapse-free syntactic theories exist with undecidable unification problems. It is also undecidable whether a given finite set E of collapse-free identities is resolvent or whether the theory defined by E is syntactic. The standard A-unification algorithm by mutation coincides with the algorithm of Plotkin (1972). Mutation together with decomposition and merging need not terminate, but since we will consider only singleton unification problems and our terms are linear, we do not need merging.[4]

It is not clear from the literature whether the combination of mutation rules for shallow and collapsing axioms and those for syntactic axioms automatically bear a complete procedure. The next section outlines a proof for the completeness of the combination of right identity and associativity, without relying on the notion of syntactic-ness.

Without exploiting the following in this thesis, we state two more results useful for theories of path logics.

(i) Kirchner and Klay (1990) prove every collapse-free and finitary unifying theory is syntactic. With the exception of the theories for T and B all equations of Figures 2.7, 2.8 and 3.4 are non-collapsing.

(ii) Doggaz and Kirchner (1991) present a completion algorithm for automatically converting a set E of linear and collapse-free equations to a finite resolvent set of equations.

[4] I do not know whether there are syntactic theories with undecidable unification problems even without merging.

5.7 Mutation for T and 4

The normalising functions N_T and N_4 rearrange paths according to the rewrite rules

$$[x\underline{e}] \Rightarrow x \quad \text{and}$$
$$[x(\alpha \circ \beta)] \Rightarrow [x\alpha\beta].$$

Inductive specifications of N_T and N_4 from paths to paths are:

$$\begin{aligned} N_T([]) &= [], \\ N_T([s\underline{e}]) &= N_T(s), \quad \text{and} \\ N_4([]) &= [], \\ N_4([su]) &= [N_4(s)u] \quad \text{provided } u \text{ is a functional variable or constant, and} \\ N_4([s(v \circ v')] &= N_4([N_4([sv])v']). \end{aligned}$$

They combine in accordance with $N_{T4}(s) = N_T(N_4(s))$, which first eliminates the $\circ$ operation and then the identity constant $\underline{e}$. Any term $N_{T4}(s)$ is said to be in *S4-normal form.* Clearly, computing the normal forms $N_T(s)$, $N_4(s)$ and $N_{T4}(s)$ of any finite path s is decidable. The set of unification rules for normalised terms is smaller and it is most likely that unification is more efficient.

We will focus on E-unification of a single equation. This is not a restriction, as general E-unification can be solved by calls of E-unification of single equations, and in calculi employing binary resolution and syntactic factoring E-unification of singleton problems suffices. More specifically, we make the following assumption.

> Any unification problem has the form $P = \{s =^? t\}$ where s and t are variable disjoint basic paths, (i) $\{s, t\}$ is prefix stable, (ii) s and t do not contain world variables, and (iii) are normalised by N_{T4}.

(ii) is ensured since $\neg\Upsilon\Pi_f(\varphi)$ contains no world variables and no world variables will be introduced during unification. Therefore, *admissible* substitutions have the form

$$\alpha \mapsto u \quad \text{with } u \text{ a functional term.}$$

Our unification rules for right identity and associativity are those listed in Figure 5.4.[5] s and t may denote empty paths, except where stated otherwise. $=^?$ is assumed to be unoriented. Observe, the system does not decompose or mutate functional terms involving $\circ$, and no normalisation is done in the unification algorithm. The rules are (in essence) instances of a subset of the rules from Comon et al. (1994).[6]

There is a subtle difference of Mutate-T and Mutate-4 to the respective instances of the general mutation rules, which are

$$\begin{aligned} P \cup \{[su] =^? t\} \quad &\rightsquigarrow \quad P \cup \{s =^? t,\ u =^? \underline{e}\} \quad \text{and} \\ P \cup \{[su] =^? [tv]\} \quad &\rightsquigarrow \quad P \cup \{[s\alpha'] =^? t,\ u =^? \alpha' \circ v\}. \end{aligned}$$

[5] The rule I named Variable Eliminate is referred to as Quantifier Eliminate in presentations, like Jouannaud and Kirchner (1991), where new variables are indicated by existential quantifiers.

[6] For readers familiar with this paper we note in our context their cycle breaking rule can be easily seen to be superfluous, since there are no world variables in the original problem, and for the functional variables Cycle applies to equations of the form $\alpha \circ u =^? \alpha$ or $u \circ \alpha =^? \alpha$, which our algorithm does not produce as we will see.

$$
\begin{array}{llll}
\textbf{Delete} & P \cup \{s =^? t\} & \rightsquigarrow & P \\
 & P \cup \{u =^? u\} & \rightsquigarrow & P \\
\textbf{Variable Elim.} & P \cup \{\alpha =^? u\} & \rightsquigarrow & P\{\alpha \mapsto u\} \\
 & \multicolumn{3}{l}{\text{provided } \alpha \text{ is an introduced variable and does not occur in } u.} \\
\textbf{Decompose} & P \cup \{[su] =^? [tv]\} & \rightsquigarrow & P \cup \{s =^? t,\ \ u =^? v\} \\
\textbf{Mutate-}T & P \cup \{[s\alpha] =^? t\} & \rightsquigarrow & P \cup \{s =^? t, \alpha =^? \underline{e}\} \\
\textbf{Mutate-}4 & P \cup \{[s\alpha] =^? [tv]\} & \rightsquigarrow & P \cup \{[s\alpha'] =^? t,\ \ \alpha =^? \alpha' \circ v\} \\
 & \multicolumn{3}{l}{\text{provided } \alpha' \text{ is a new variable and not both } s \text{ and } t \text{ are empty.}}
\end{array}
$$

Figure 5.4: Unification rules for the path logics associated with T and 4

It is clear that the case when u is a constant different from $\underline{e}$ in the first rule leads to an unsatisfiable situation $\underline{\alpha} =^? \underline{e}$. In the second rule when u is a constant this leads to the situation $\underline{\alpha} =^? \alpha' \circ v$ which is unsatisfiable when v is not logically equivalent to $\underline{e}$, and in this case $\underline{\alpha} =^? \alpha'$ is redundant.

Mutate-T binds a variable in a right-most position with the identity constant $\underline{e}$ and deletes the variable from the original term. For example, the formula $\Box\Diamond p \to \Box p$ is a theorem in KT and translates to the input set $\{P[\alpha\underline{\beta}],\ \ \neg P[\gamma]\}$. We want to find a complete set of unifiers and if possible a minimal complete set of unifiers, for

$$\{[\alpha\underline{\beta}] =^? [\gamma]\}.$$

The only unifier is $\{\alpha \mapsto \underline{e}, \gamma \mapsto \underline{\beta}\}$. The empty clause follows. For the formula $\Diamond(\Box p \to \Box p)$, which is also a theorem in KT, with input set $\{P[\alpha\beta],\ \ \neg P[\gamma]\}$ the unification problem

$$\{[\alpha\beta] =^? [\gamma]\}$$

has two minimal (most general) unifiers:

$$\{\alpha \mapsto \gamma,\ \ \beta \mapsto \underline{e}\} \quad \text{and} \quad \{\alpha \mapsto \underline{e},\ \ \beta \mapsto \gamma\}.$$

The algorithm computes a third unifier, namely $\{\alpha \mapsto \underline{e},\ \ \beta \mapsto \underline{e},\ \ \gamma \mapsto \underline{e}\}$, which is not most general.

Mutate-4 applies to terms $s = [u_0 u_1 \dots u_m] =^? [v_0 v_1 \dots v_n] = t$ with the pair (u_m, v_n) either being a variable-constant, a constant-variable or a variable-variable pair. For the first two constellations there is one transformation by Mutate-4 and for the last constellation there are two.

$$
\begin{array}{ll}
\multicolumn{2}{l}{\{[u_0 u_1 \dots u_{m-1}\alpha] =^? [v_0 v_1 \dots v_{n-1}\underline{\beta}]\}} \\
\overset{4}{\rightsquigarrow} & \{\alpha =^? \alpha' \circ \underline{\beta},\ \ [u_0 u_1 \dots u_{m-1}\alpha'] =^? [v_0 v_1 \dots v_{n-1}]\} \\
\multicolumn{2}{l}{\{[u_0 u_1 \dots u_{m-1}\alpha] =^? [v_0 v_1 \dots v_{n-1}\gamma]\}} \\
\overset{4}{\rightsquigarrow} & \{\alpha =^? \alpha' \circ \gamma,\ \ [u_0 u_1 \dots u_{m-1}\alpha'] =^? [v_0 v_1 \dots v_{n-1}]\} \\
\text{or} & \{\gamma =^? \gamma' \circ \gamma,\ \ [u_0 u_1 \dots u_{m-1}] =^? [v_0 v_1 \dots v_{n-1}\gamma']\}.
\end{array}
$$

Provided we add appropriate deletion rules, the search tree for transformations with Mutate-4 is an instance of the search tree of Plotkin's (1972) algorithm for word unification modulo associativity (applied to paths and employing the right-to-left as opposed to the left-to-right strategy).

Compare the derivation according to Ohlbach's method in Figure 5.3 with the derivation in Figure 5.5 according to the mutation system. The successful branches in the derivation tree yield the following unifiers:

1. $\{\delta \mapsto \underline{\gamma},\ \ \beta \mapsto \underline{\beta} \circ \underline{\gamma},\ \ \alpha \mapsto \underline{\alpha}\}$
2. $\{\delta \mapsto \underline{\gamma},\ \ \beta \mapsto (\beta'' \circ \underline{\beta}) \circ \underline{\gamma},\ \ \alpha \mapsto \underline{\alpha} \circ \beta''\}$
3. $\{\delta \mapsto \underline{e},\ \ \beta \mapsto \underline{\beta},\ \ \alpha \mapsto \underline{\alpha}\}$
4. $\{\delta \mapsto \underline{e},\ \ \beta \mapsto \beta' \circ \underline{\beta},\ \ \alpha \mapsto \underline{\alpha} \circ \beta'\}$
5. $\{\delta \mapsto \delta' \circ \underline{\gamma},\ \ \beta \mapsto (\underline{\beta} \circ \underline{\gamma}) \circ \delta',\ \ \alpha \mapsto \underline{\alpha}\}$
6. $\{\delta \mapsto \delta' \circ \underline{\gamma},\ \ \beta \mapsto ((\beta'' \circ \underline{\beta}) \circ \underline{\gamma}) \circ \delta',\ \ \alpha \mapsto \underline{\alpha} \circ \beta''\}$.

Clearly, the search tree is considerably smaller and there are no repetitions in the solution set. The solution set is not minimal, though.

Now, we prove our algorithm is sound and complete. By definition, a set of transformation rules is *complete* in a theory E if the following two conditions hold:

Soundness: If P transforms to P' by the application of any of the transformation rules, written $P \overset{*}{\leadsto} P'$, then every E-unifier of P' is an E-unifier of P.

Completeness: For any E-unifier θ of P, there is some P' in solved form such that $P \overset{*}{\leadsto} P'$ and the idempotent unifier σ associated with P' is more general than θ with respect to the variables occurring in P, written $\sigma \leq_E \theta[\mathit{Var}(P)]$.

Formally, a *solved form* is either the empty set or a finite set of the form $\{\alpha_1 =^? u_1,\ \ldots,\ \alpha_n =^? u_n\}$ and $\alpha_1, \ldots, \alpha_n$ are distinct variables occurring in no u_i. A variable α is *solved* in a set P if P includes an $\alpha =^? u$ (or $u =^? \alpha$) and α occurs exactly once in P. A variable that is not solved is an *unsolved variable.* By definition, $\sigma =_E \theta[V]$ iff for any variable x in V, $x\sigma$ and $x\theta$ are E-equivalent, and $\sigma \leq_E \theta[V]$ iff there is a substitution such that $\sigma\sigma' =_E \theta[V]$.

Equivalence (inequivalence and inclusion) modulo right identity and associativity will be denoted by $=_{S4}$ ($\neq_{S4}$ and $\leq_{S4}$).

Theorem 5.7.1 The system of Figure 5.4 is sound.

Proof. By inspecting the rules. □

In the remainder of this section, P denotes a singleton unification problem of basic paths satisfying assumptions (i), (ii) and (iii). Let P' be obtained from P by any sequence of transformations modulo the rules of Figure 5.4. For the next lemma (and also Lemma 5.7.7) it is important that the initial pair in P is variable disjoint.

Lemma 5.7.2 For any identity $\alpha =^? u$ in P', the variable α does not occur in u.

Proof. By inspecting the rules. □

$$
\begin{array}{l}
[\alpha\underline{\beta}\underline{\gamma}\delta] =^{?} [\underline{\alpha}\beta\underline{\gamma}] \\
\quad \overset{\text{Dec.}}{\rightsquigarrow} \quad \delta =^{?} \underline{\gamma},\ \ [\alpha\underline{\beta}\underline{\gamma}] =^{?} [\underline{\alpha}\beta] \\
\qquad\qquad\qquad \left.\begin{array}{l}
\overset{\text{Dec.}}{\rightsquigarrow} \quad \beta =^{?} \underline{\gamma},\ \ [\alpha\underline{\beta}] =^{?} [\underline{\alpha}] \quad \text{not solved} \\
\overset{T}{\rightsquigarrow} \quad \beta =^{?} \underline{e},\ \ [\alpha\underline{\beta}\underline{\gamma}] =^{?} [\underline{\alpha}] \quad \text{not solved} \\
\overset{4}{\rightsquigarrow} \quad \beta =^{?} \beta' \circ \underline{\gamma},\ \ [\alpha\underline{\beta}] =^{?} [\underline{\alpha}\beta'] \\
\qquad\qquad\qquad \overset{2\times\text{Dec.}}{\rightsquigarrow} \quad \beta' =^{?} \underline{\beta},\ \ \alpha =^{?} \underline{\alpha} \quad \rightsquigarrow \quad 1. \\
\qquad\qquad\qquad \overset{T}{\rightsquigarrow} \quad \beta' =^{?} \underline{e},\ \ [\alpha\underline{\beta}] =^{?} [\underline{\alpha}] \quad \text{not solved} \\
\qquad\qquad\qquad \overset{2\times 4}{\rightsquigarrow} \quad \beta' =^{?} \beta'' \circ \underline{\beta},\ \ \alpha =^{?} \underline{\alpha} \circ \beta'' \quad \rightsquigarrow \quad 2.
\end{array}\right\} (*) \\
\quad \overset{T}{\rightsquigarrow} \quad \delta =^{?} \underline{e},\ \ [\alpha\underline{\beta}\underline{\gamma}] =^{?} [\underline{\alpha}\beta\underline{\gamma}] \\
\qquad\qquad\qquad \overset{\text{Dec,Del}}{\rightsquigarrow} \quad [\alpha\underline{\beta}] =^{?} [\underline{\alpha}\beta] \\
\qquad\qquad\qquad\qquad\qquad \overset{2\times\text{Dec}}{\rightsquigarrow} \quad \beta =^{?} \underline{\beta},\ \ \alpha =^{?} \underline{\alpha} \quad \rightsquigarrow \quad 3. \\
\qquad\qquad\qquad\qquad\qquad \overset{T}{\rightsquigarrow} \quad \beta =^{?} \underline{e},\ \ [\alpha\underline{\beta}] =^{?} [\underline{\alpha}] \quad \text{not solved} \\
\qquad\qquad\qquad\qquad\qquad \overset{2\times 4}{\rightsquigarrow} \quad \beta =^{?} \beta' \circ \underline{\beta},\ \ \alpha =^{?} \underline{\alpha} \circ \beta' \quad \rightsquigarrow \quad 4. \\
\quad \overset{4}{\rightsquigarrow} \quad \delta =^{?} \delta' \circ \underline{\gamma},\ \ [\alpha\underline{\beta}\underline{\gamma}\delta'] =^{?} [\underline{\alpha}\beta] \\
\qquad\qquad\qquad \overset{\text{Dec}}{\rightsquigarrow} \quad \beta =^{?} \delta',\ \ [\alpha\underline{\beta}\underline{\gamma}] =^{?} [\underline{\alpha}] \quad \text{not solved} \\
\qquad\qquad\qquad \overset{T}{\rightsquigarrow} \quad \delta' =^{?} \underline{e},\ \ [\alpha\underline{\beta}\underline{\gamma} =^{?} [\underline{\alpha}\beta] \quad \rightsquigarrow \quad \ldots \qquad \text{not solved} \\
\qquad\qquad\qquad \overset{T}{\rightsquigarrow} \quad \beta =^{?} \underline{e},\ \ \ldots \quad \text{not solved} \\
\qquad\qquad\qquad \overset{4}{\rightsquigarrow} \quad \beta =^{?} \beta' \circ \delta',\ \ [\alpha\underline{\beta}\underline{\gamma}] =^{?} [\underline{\alpha}\beta] \\
\qquad\qquad\qquad\qquad\qquad\qquad \vdots\ \text{like } (*) \quad \rightsquigarrow \quad 5.\ \text{and}\ 6.
\end{array}
$$

Figure 5.5: A sample derivation of *S4*-unifiers

Lemma 5.7.3 Every P' irreducible by the rules of Figure 5.4 is in solved form or it is unsatisfiable in $=_{S4}$.

Proof. The only irreducible equations not in the form $\alpha =^? u$ with α a solved variable, have the form: $[s\underline{\alpha}] = []$, $\underline{\alpha} =^? \underline{\beta}$ and $\underline{\alpha} =^? \underline{e}$. If P' includes any one of these it is not solvable. Observe, as $\underline{e}$ does not occur in the starting P situations like $[s\underline{e}t] =^? []$ do not arise. □

Theorem 5.7.4 The system of Figure 5.4 is complete.

The core structure of the proof is standard. We let

$$P = \{[su] =^? [tv]\}$$

with $s = [s|_1 \dots s|_m]$ and $t = [t|_1 \dots t|_n]$, each non-empty, that is, $1 \leq m, n$. We let θ be any $S4$-unifier of P, that is,

$$[su]\theta =_{S4} [tv]\theta.$$

The aim is to show there is a sequence of transformation of P such that the associated unifier σ is more general than θ. In parallel to transforming P we extend the unifier θ by adding bindings of new variables to θ obtaining θ'. Below, in Lemmas 5.7.8 and 5.7.9, we will define θ' in such a way that if θ unifies P and $P \rightsquigarrow P'$, that is, P transforms to P' in one step, then θ' unifies P'. The resulting procedure starts with the pair (P, θ) and computes at least one pair (P', θ'), such that

(i) $P \overset{*}{\rightsquigarrow} P'$,

(ii) P' is in solved form,

(iii) $\theta \subseteq \theta'$ and $\theta'|_{Var(\theta)} = \theta$ (the restriction of θ' to the variables of θ coincides with θ).

By assumption θ is a unifier of P, and consequently, by induction on the proof length, θ' of the final pair is a unifier of P'. This establishes completeness, when every derivation is finite.

The next lemmas supply the missing details.

Lemma 5.7.5 The unifier σ associated with the solved P' is more general than θ' with respect to the variables of P'.

Proof. Every equation in P' has the form $\alpha =^? u$. Since θ' unifies P', $\alpha\theta' = u\theta'$. σ is also a unifier of P', hence $\alpha\sigma = u\sigma = u$. Therefore, for any variable α in P'

$$\alpha\theta' = \alpha\sigma\theta'$$

and this means σ is more general than θ' with respect to $Var(P')$. □

Since θ and θ' restricted to the variables of P are equivalent, it follows that:

Lemma 5.7.6 The unifier σ associated with P' is more general than θ with respect to the variables of P' and P.

The procedure of transforming (P, θ) to a suitable (P', θ') terminates:

Lemma 5.7.7 Any fair implementation of a unification algorithm for the transformation system of Figure 5.4 terminates for any P satisfying the assumptions.

Proof. Let $\tau(s)$ denote the functional depth of a term s. Define a measure μ of any unification problem P by $\mu(P) = (d, v)$, where v denotes the number of unsolved variables in P, and d is determined by the depths of the pairs of world terms in P (of which there are at most one). More specifically, $d = \tau(s) + \tau(t)$ if $s =^? t \in P$ and both s and t are of type world, and $d = 0$ if no such pair exists.

Examine each transformation rule in turn to see that $\mu(P')$ is smaller than $\mu(P)$ under the lexicographical ordering. Except for Variable Eliminate each rule decreases the value of d. Variable Eliminate leaves d unchanged but it decreases v. The rules do not convert the status of any variable from solved to unsolved. □

Next, we consider one step conversions of any pair (P, θ) to a suitable pair (P', θ') (in two lemmas). In order to avoid cluttering in the proofs we write just '$=$' in place of '$=_{S4}$'.

Lemma 5.7.8 Consider $P \cup \{[su] =^? [tv]\}$ with $s = [s|_0 \dots s|_m]$ and $t = [t|_0 \dots t|_n]$ for $1 \leq m, n$. The terms $[su]$ and $[tv]$ are assumed to be in *S4*-normal form. Let θ be any *S4*-unifier of $P \cup \{[su] =^? [tv]\}$, in particular,

$$[su]\theta =_{S4} [tv]\theta. \tag{5.3}$$

θ' as defined in the following is in each case an *S4*-unifier of P'.

(i) If $u\theta =_{S4} v\theta$, then let $\theta' = \theta$ and apply Decompose to P, yielding

$$P' = P \cup \{s =^? t,\ \ u =^? v\}.$$

(ii) If $u\theta =_{S4} \underline{e}$, then let $\theta' = \theta$ and apply Mutate-T to P, yielding

$$P' = P \cup \{s =^? [tv],\ \ u =^? \underline{e}\}.$$

(iii) If u is a variable α, say, and $u\theta =_{S4} [t|_k \dots t|_n v]\theta$ for some $1 \leq k \leq n$, then let

$$\theta' = \theta_0\theta \quad \text{with } \theta_0 = \{\alpha' \mapsto [t|_k \dots t|_n]\}$$

and apply Mutate-4 to P, yielding

$$P' = P \cup \{[s\alpha'] =^? t,\ \ \alpha =^? \alpha' \circ v\},$$

for α' a new variable not occurring in P or θ.

Proof. (i) By assumption (5.3), $[su]\theta = [tv]\theta$, which implies $[(s\theta)(u\theta)] = [(t\theta)(v\theta)]$. Then, since $u\theta = v\theta$,

$$[(s\theta)(u\theta)] = [(t\theta)(u\theta)].$$

Consequently, $s\theta = t\theta$. Therefore, $\theta = \theta'$ is an *S4*-unifier of P'.

(ii) $\theta = \theta'$ is an $S4$-unifier of P', since $u\theta = \underline{e}$ and

$$\begin{aligned}[tv]\theta &= [su]\theta \qquad \text{by (5.3)}\\ &= [(s\theta)(u\theta)] = [(s\theta)\underline{e}] \qquad \text{since } u\theta = \underline{e}\\ &= s\theta \qquad \text{by the right identity law.}\end{aligned}$$

(iii) θ' is well-defined since α' is a new variable that does not occur in α.
(a) θ' is an $S4$-unifier of $\alpha =^? \alpha' \circ v$:

$$\begin{aligned}(\alpha' \circ v)\theta' &= (\alpha' \circ v)\theta_0\theta\\ &= ([t|_k \ldots t|_n] \circ v)\theta \qquad \text{since } \theta_0 = \{\alpha' \mapsto [t|_k \ldots t|_n]\}\\ &= \alpha\theta \qquad \text{since } \alpha\theta = [t|_k \ldots t|_n v]\theta\\ &\qquad\qquad\qquad\qquad = ([t|_k \ldots t|_n] \circ v)\theta\\ &= \alpha\theta_0\theta = \alpha\theta' \qquad \text{since } \alpha\theta_0 = \alpha.\end{aligned}$$

(b) θ' is an $S4$-unifier of $[s\alpha'] =^? t$:

$$\begin{aligned}[s\alpha]\theta' = [s\alpha]\theta_0\theta = [s\alpha]\theta &= [tv]\theta \qquad \text{by (5.3)}\\ &= [tv]\theta_0\theta = [tv]\theta'.\end{aligned}$$

That is, $[s\alpha]\theta' = [tv]\theta'$, which implies

$$[(s\theta')(\alpha\theta')] = [([t|_0 \ldots t|_{k-1}]\theta')([t|_k \ldots t|_n v]\theta')].$$

Since $[t|_k \ldots t|_n v]\theta' = [t|_k \ldots t|_n v]\theta_0\theta = [t|_k \ldots t|_n v]\theta = \alpha\theta$ by (a),

$$[(s\theta')(\alpha\theta')] = [([t|_0 \ldots t|_{k-1}]\theta')(\alpha\theta')].$$

Therefore, $s\theta' = [t|_0 \ldots t|_{k-1}]\theta'$. Then

$$[s\alpha']\theta' = [(s\theta')(\alpha'\theta')] = [([t|_0 \ldots t|_{k-1}]\theta')(\alpha'\theta')].$$

$\alpha'\theta' = \alpha'\theta_0\theta = [t|_k \ldots t|_n]\theta = [t|_k \ldots t|_n]\theta_0\theta = [t|_k \ldots t|_n]\theta'$. Hence

$$\begin{aligned}[s\alpha']\theta' &= [([t|_0 \ldots t|_{k-1}]\theta')(\alpha'\theta')]\\ &= [([t|_0 \ldots t|_{k-1}]\theta')([t|_k \ldots t|_n]\theta')]\\ &= [t|_0 \ldots t|_{k-1} t|_k \ldots t|_n]\theta'\\ &= t\theta'.\end{aligned}$$

This means, θ' is an $S4$-unifier of $[s\alpha'] =^? t$. □

The lemma also covers the cases that $v\theta =_{S4} \underline{e}$ and $v\theta =_{S4} [s|_k \ldots s|_n]\theta$ for some $1 \le k \le m$ and v a variable. Observe that when $u\theta =_{S4} [t|_k \ldots t|_n v]\theta$ but both u and v are constants, the conditions of either (i) or (ii) hold. If u and v are both constants then either (a) $u = v$ or (b) $u = \underline{\alpha}$, say, and $v = \underline{e}$. (a) implies $u\theta = v\theta$, and (b) implies $v\theta = \underline{e}$.

It remains to clarify whether there are cases that the lemma does not cover. Is there a case such that neither of the following hold?

(i) $u\theta =_{S4} v\theta$,

(ii) $u\theta =_{S4} \underline{e}$ (or $v\theta =_{S4} \underline{e}$),

(iii) u is a variable and $u\theta =_{S4} [t|_k \ldots t|_n v]\theta$ for some $1 \le k \le n$ (or v is a variable and $v\theta =_{S4} [s|_k \ldots s|_n]\theta$ for some $1 \le k \le m$).

The answer is, yes, as in this example

$$P = \{[\underline{\alpha}\underline{\alpha}'\beta] =^? [\underline{\alpha}\delta\underline{\gamma}]\} \quad \text{and} \quad \theta = \{\beta \mapsto \underline{\epsilon} \circ \underline{\gamma},\ \delta \mapsto \underline{\alpha}' \circ \underline{\epsilon}\}$$

when in the general case

$$[s|_k \ldots s|_m u]\theta =_{S4} [t|_l \ldots t|_n v]\theta$$

is true, for some $1 \le k \le m$ and $1 \le l \le n$. If u and v are both constants then, as above, either $u\theta = v\theta$ or $u\theta = \underline{e}$ or $v\theta = \underline{e}$. The next result deals with the case that one of u or v is a variable.

Lemma 5.7.9 Let θ be an *S4*-unifier of $P \cup \{[s\alpha] =^? [tv]\}$ with $s = [s|_0 \ldots s|_m]$ and $t = [t|_0 \ldots t|_n]$ for $1 \le m, n$, and both $[s\alpha]$ and $[tv]$ are in *S4*-normal form. Let

$$[s|_k \ldots s|_m \alpha]\theta =_{S4} [t|_l \ldots t|_n v]\theta$$

for some $1 \le k \le m$ and $1 \le l \le n$. If θ includes a binding of α to u, that is, $\alpha \mapsto u \in \theta$, then let

$$\theta' = \theta\theta_0 \quad \text{with } \theta_0 = \{\alpha' \mapsto u'\}$$

where u' is given by $u =_4 u' \circ v'$ and $v' = v\theta$, and apply Mutate-4 to P, yielding

$$P' = P \cup \{[s\alpha'] =^? t,\ \ \alpha =^? \alpha' \circ v\},$$

for α' a new variable not occurring in P or θ. Then θ' unifies P'.

Proof. (a) θ' is an *S4*-unifier of $\alpha =^? \alpha' \circ v$, since

$$(\alpha' \circ v)\theta' = (\alpha'\theta_0) \circ (v'\theta_0) = u' \circ v' = u = \alpha\theta = \alpha\theta'.$$

(b) θ' is an *S4*-unifier of $[s\alpha'] =^? t$: By assumption $[s\alpha]\theta = [tv]\theta$, hence $[(s\theta)(\alpha\theta)] = [(t\theta)(v\theta)]$. As $\alpha\theta = u = u' \circ v'$ and $v\theta = v'$ we have

$$[(s\theta)(u' \circ v') = [(s\theta)a'v'] = [(t\theta)v'].$$

It follows that $[(s\theta)a'] = t\theta$. Then, $[s\alpha]\theta' = [s\alpha]\theta\theta_0 = [s\alpha]\theta = t\theta = t\theta'$, as required. □

For example, the sample pair

$$P = \{[\underline{\alpha}\underline{\alpha}'\beta] =^? [\underline{\alpha}\delta\underline{\gamma}]\} \quad \text{and} \quad \theta = \{\beta \mapsto \underline{\epsilon} \circ \underline{\gamma},\ \delta \mapsto \underline{\alpha}' \circ \underline{\epsilon}\}$$

is converted to

$$P' = \{[\underline{\alpha}\underline{\alpha}'\beta'] =^? [\underline{\alpha}\delta],\ \beta =^? \beta' \circ \underline{\gamma}\} \quad \text{and} \quad \theta' = \{\beta \mapsto \underline{\epsilon} \circ \underline{\gamma},\ \delta \mapsto \underline{\alpha}' \circ \underline{\epsilon},\ \beta' \mapsto \underline{\epsilon}\}.$$

The lemma makes assumptions, that are not met in the following two situations. First, if no u' exists such that $u =_4 u' \circ (v\theta)$ then $v\theta$ is equivalent to $\underline{e}$. This case is dealt with in (ii) of the previous lemma. Second, the situation that neither α nor v are in the domain of θ and $\alpha\theta \neq_{S4} v\theta$ is impossible (for otherwise $[s\alpha]$ and $[tv]$ are not unifiable).

Observe, the lemma implies (iii) of the previous lemma. This completes the proof of Theorem 5.7.4.

5.8 Preservation of prefix stability

Finally we verify that the application of *S4*-unifiers and *S4*-resolution preserves prefix stability. This justifies the assumptions made in the previous section, namely, that the terms in the initial problem set are basic paths and the world terms on the left hand sides of the transformation rules of Figure 5.4 are also basic paths. We also prove a preservation result for forming $\mathcal{R}^{S4}_{N_{T4}}$ and $\mathcal{R}^{S4}_{\text{COND}\,\circ\,N_{T4}}$-resolvents. The proofs are very similar to those for applying syntactic unifiers and forming standard resolvents (Section 5.4).

In the following we let T be a set of terms in the vocabulary of basic path logic, because remember, every theory resolvent is immediately normalised by N_{T4}. The analogue of Theorem 5.4.1 is not true in its full generality for bindings of *S4*-unifiers. It is true when suffixes are variable disjoint, and when more restrictions (to be made precise below) hold for instantiations with $\circ$ terms. For bindings of the form $\alpha \mapsto \underline{e}$ the following is immediate by Theorem 5.4.1.[7]

Corollary 5.8.1 Let T be a set of terms in the vocabulary of basic path logic. Let $s = [u_0 u_1 \dots u_m]$ and $t = [v_0 v_1 \dots v_n]$ be two terms in T such that for some $k > 0$,

$$u_0 = v_0, u_1 = v_1, \dots, u_{k-1} = v_{k-1} \quad \text{and} \quad u_k \neq v_k,$$

u_k is a variable, and the suffixes $[u_{k+1} \dots u_m]$ and $[v_{k+1} \dots v_n]$ are variable disjoint. Let σ be a substitution $\{u_k \mapsto \underline{e}\}$. Then $N_T(T\sigma)$ satisfies T1 and T2, provided T does.

For bindings of the form $u_k \mapsto v \circ v'$, because they cause the term depth to increase, the concept of k-equality needs to be generalised to the concept of (k,l)-equality. Two basic paths s and t are *(k,l)-equal* if t is like s except possibly the term $s|_k$ in the k-th position is replaced by a string $w_1 \dots w_l$ of length l. In other words, s and t are (k,l)-equal provided $s = t$, or when $s = [s|_1 \dots s|_m]$ then $t = [s|_1 \dots s|_{k-1} w_1 \dots w_l s|_{k+1} \dots s|_m]$, or the other way around.

Theorem 5.8.2 Let s and t be two terms in T defined as in the previous result. Let σ be a substitution $\{u_k \mapsto w\}$ where

$$N_{T4}([w]) = [w_1 \dots w_l] \quad \text{and} \quad w_1 = v_k,$$

and s and w are variable disjoint.[8] Then $N_{T4}(T\sigma)$ satisfies T1 and T2, provided T, $N_{T4}([w])$ and the set $\{[w], [v_k \dots v_n]\}$ do.

Proof. Proceed as in the proof of Theorem 5.4.1 with the obvious modifications. Let s and t be any terms in T that satisfy the conditions T1 and T2 and consider $N_{T4}(s\sigma)$ and $N_{T4}(t\sigma)$ in $N_{T4}(T\sigma)$. It is not difficult to verify that the pairs $N_{T4}(s)$ and $N_{T4}(s\sigma)$, and also $N_{T4}(t)$ and $N_{T4}(t\sigma)$, are (k,l)-equal. We assume without loss of generality that $s|_k = u_k$ (for, otherwise if u_k does not occur in either of s or t then the result is trivially true). Then

$$N_{T4}(s\sigma) = [s|_1 \dots s|_{k-1} w_1 \dots w_l s|_{k+1} \dots s|_m].$$

[7] This is also true for bindings produced by other collapsing equations provided there are no interactions with other operations preventing elimination by normalisation.

[8] More accurately, $N_{T4}([w])$ coincides with $N_{T4}([[], w]) = [[] w_1 \dots w_l]$.

Since s and $[w_1 \dots w_l]$ have no common variables and w satisfies T1 and T2, so does $s\sigma$.

Now, consider two cases: (i) σ leaves t unchanged so that $t\sigma = t$ and (ii) it does not. In the either case we need to prove T1 and T2 hold for i and j strictly below $k + l$. This is tedious and as the arguments are similar to those of Theorem 5.4.1, we omit the details. □

This theorem and the previous corollary will be used in the induction argument over the decomposition into bindings of idempotent unifiers proving preservation of T1 and T2 (Theorem 5.8.4).

Lemma 5.8.3 Let s and t be two variable disjoint basic paths. Let σ be any *S4*-unifier computed by the system of Figure 5.4. Then

(i) σ is an idempotent unifier.

(ii) $\sigma = \sigma_1 \dots \sigma_l$, where the σ_i are of the form $\{\alpha_i \mapsto w\}$ such that for any pair σ_i and σ_j with $1 \leq i < j \leq n$, if α_i and α_j occur at positions k_i and k_j in s or t, then $k_i \leq k_j$.

(iii) If α_1 of σ_1 is a variable occurring in s then the following are equivalent.

a. $s = [u_0 u_1 \dots u_m]$ and $t = [v_0 v_1 \dots v_n]$ have a common prefix $[u_0 u_1 \dots u_{k+1}]$, $u_k \neq v_k$ and u_k is a variable.

b. Either $\sigma_1 = \{u_k \mapsto \underline{e}\}$ or $\sigma_1 = \{u_k \mapsto w\}$ where $N_4([w]) = [w_1 \dots w_l]$ and $w_1 = v_k$.

(iv) $N_4([w])$ satisfies T1 and T2 provided s and t do.

(v) $\{N_4([w]), [v_k \dots v_n]\}$ satisfies T1 and T2 provided s and t do.

Proof. (i) is evident by the definition of solved forms.

Consequently, σ coincides with a composition of bindings σ_i. Compose the bindings as determined by the positions in s and t of the variables α_i. This verifies (ii).

(iii) is true, for, otherwise s and t are not unifiable.

(iv) No unification rule duplicates variables, hence $N_4([w])$ is a linear term and satisfies T1 and T2.

(v) *S4*-bindings of the form $\alpha_i \mapsto w$ are such that: $N_4([w])$ has length smaller or equal to $[v_k \dots v_n]$, and $w_1 = v_k$, $w_2 = v_{k+1}$, ... , $w_{l-1} = v_{k+l-2}$. If $w_l = v_{k+l-1}$ then $N_4([w])$ is a prefix of $[v_k \dots v_n]$, which means $\{N_4([w]), [v_k \dots v_n]\}$ satisfies T1 and T2. If $w_l \neq v_{k+l-1}$ then w_l is a variable introduced by an application of Mutate-4, v_{k+l-1} is a variable and $\sigma_2 = \{v_{k+l-1} \mapsto w'\}$. Also, in this case $\{N_4([w]), [v_k \dots v_n]\}$ satisfies T1 and T2. □

Theorem 5.8.4 Let σ be an *S4*-unifier of two variable disjoint terms s and t in T. If T satisfies properties T1 and T2 then so does $N_{T4}(T\sigma)$.

Proof. Let σ be $\sigma_1 \dots \sigma_l$ as in (ii) of the previous lemma. Iteratively, consider the triples s, t and σ_1, then $N_{T4}(s\sigma_1)$, $N_{T4}(t\sigma_1)$ and σ_2, etcetera, and apply Corollary 5.8.1 and Theorem 5.8.2. By (iii), (iv) and (v) of the previous lemma, in any iteration the conditions T1 and T2 are satisfied by any $N_{T4}(s\sigma_1 \dots \sigma_i)$, $N_{T4}(t\sigma_1 \dots \sigma_i)$ and σ_{i+1}. □

Consequently, using Theorem 5.4.6 which says that the union of two variable disjoint sets of prefix stable terms is prefix stable, the preservation result for binary theory resolvents follows. More generally:

Theorem 5.8.5 For $E \subseteq \{\text{right identity}, \text{associativity}\}$, the binary $\mathcal{R}^E_{N_E}$-resolvent of two variable disjoint clauses satisfying T1 and T2 also satisfies T1 and T2.

The main preservation theorem follows:

Theorem 5.8.6 Let $E \subseteq \{\text{right identity}, \text{associativity}\}$, and let φ be any modal formula and $S = c(\neg\Upsilon\Pi_f(\varphi))$. For any n, $(\mathcal{R}^E_{N_E})^n(S)$ and $(\mathcal{R}^E_{\text{COND} \,\circ\, N_E})^n(S)$ are well-formed in the basic path logic.

Proof. By the previous theorem and by preservation of prefix stability under syntactic factoring and condensing. □

5.9 Conclusion

In summary, this chapter defines a lattice of path logics and describes resolution and unification procedures for basic path logic and for extensions that correspond to modal logics in which T and 4 hold, and that correspond to *S5*. It describes, for path clauses obtained by the optimised functional translation, combinations of known techniques and methods proposed for clauses obtained by the (non-optimised) functional transformation in Ohlbach (1988a, 1991), Zamov (1989), Auffray and Enjalbert (1992), and possibly other papers. One notable difference is worth stressing. Basic path clauses including normalised clauses do not contain Skolem functions other than constants. This makes exhibiting the existence of a term depth limit next to trivial, that is, with the exception of *K4* and *S4*. Thus, we have accomplished the first important goal of proving decidability of resolution with condensing, namely the existence of a term depth bound ((A) of the Preview). In general, the language restrictions are such that the proofs of the results in this chapter are somewhat simpler (though remaining tedious) than proofs for the non-optimised translations found in the current literature.

We have illustrated the search space of the unification procedure for *S4* based on mutation is considerably smaller than that of Ohlbach's procedure. The procedure is similar to the mutation-based procedure proposed by Auffray and Enjalbert (1992) for non-optimised path clauses. It paramodulates into the top position systematically reducing a pair of terms from the right to the left. And, we have proved syntactic unification can be simplified for singleton problems.

We conclude with some remarks concerning further work.

Due to the assumption we make in Section 5.7, in particular, that the input set consists of one pair of terms. For semantic factoring we need general E-unification for which our algorithm is not sufficient (this would require a deletion rule of the identity constant and a more general form of the variable elimination rule). Given a set of terms (literals), computing the syntactic most general unifier (when it exists) is easier than computing the set of minimal E-unifiers. Semantic factoring can produce an exponential number of factors causing a significant overhead. The price we pay for using syntactic factoring is incompatibility with strategies like tautology deletion. So, evidently there is a tradeoff which should be kept in mind and deserves further investigation.

Our unification algorithm is not optimal. It does not compute a minimal complete set of E-unifiers. The redundant unifiers will need to be filtered out by post processing. Possibly this can be avoided by additional unification rules similar to those of Otten and

Kreitz (1996) who present a system consisting of ten rules for terms satisfying the stronger T-string property.

Chapter 6

Decidability by unrefined resolution

The main purpose of this chapter is to prove any complete resolution procedure with condensing is a decision procedure for any finite input set of basic path clauses. We will also provide an estimate of the size of the class of variable indecomposable and condensed path clauses. This gives an indication of the complexity of unrefined resolution and condensing.

Section 6.1 outlines the fundamental idea of proving decidability. Briefly, the idea is to prove any saturated class $\mathbf{S}$ of basic path clauses produced by a finite input set is finite. We suppose a bound exists for the maximal lengths of paths in this class, and our goal is to show that a bound exists for the number of literals and/or variables in any clause of this class. This will be done by encoding clauses as matrices of terms. This so-called matrix term representation is introduced in Section 6.2. Sections 6.3 and 6.4 study variable partitions and prefix partitions of the encoding. Section 6.5 gives the decidability results for basic path logic and its generalisations and it discusses the relevance to modal logics. Section 6.6 is devoted to providing explicit bounds for the number of literals and variables of any clause of the class $\mathbf{S}$. This requires a complex embedding of variables into a finite power set algebra. The paper Schmidt (1997a) is a short version of the first four sections.

6.1 Proving decidability of resolution

The basic idea of proving decidability for resolution combined with condensing is due to Joyner Jr. (1976). By definition, a *resolution decision procedure* for an effectively specified class $\mathbf{S}$ of clauses is a complete resolution procedure $\mathcal{R}$ such that if $S \subseteq \mathbf{S}$ and S is satisfiable, then for some $n \geq 0$, $\mathcal{R}^n(S)$ and $\mathcal{R}^{n+1}(S)$ are variants.

We will focus on an arbitrary class $\mathbf{S}$ of basic path clauses formed by saturation (deductive closure) with respect to any complete resolution procedure $\mathcal{R}_{\text{COND}}$ from a finite input set S, and prove that $\mathbf{S}$ is finite. From the outset we assume the existence of a term depth bound for any clause in $\mathbf{S}$ (that is, (A) of the Preview). Our goal is to prove the existence of a clause size (cardinality) bound (that is, (B) of the Preview). Then it follows $\mathbf{S}$ is finitely bounded.

The clause size bound is tied with the maximum number of variables which can occur in any clause of $\mathbf{S}$. More precisely, it is tied with the maximum number of variables which can occur in any variable indecomposable (sub)clause. Formally, the *variable partition* of any clause C is the finest partition of C into disjoint subclauses which do not share common variables. The subclauses in the variable partition are said to be *variable indecomposable* or *split*. Different from Joyner we will not put all ground literals in C into one single block. Ground literals will belong to separate blocks, and these are singleton sets. For example, the variable partition of the clause

$$P(\underline{a}) \vee P(\underline{b}) \vee \neg Q(z') \vee \neg R(x, y) \vee \neg R(y, z) \vee R(x, z)$$

is the following.

$P(\underline{a})$	$P(\underline{b})$	$\neg Q(z')$	$\neg R(x, y) \vee \neg R(y, z) \vee R(x, z)$

Suppose there is a bound on the number of variables occurring in any variable indecomposable and condensed C of $\mathbf{S}$. Then there can only be a limited number of non-variant variable indecomposable clauses in the class. In particular, there can only be a limited number of non-variant variable indecomposable subclauses of any condensed clause. This implies that any condensed clause in the class is bounded in size. Consequently, the entire class is bounded in size.

We will develop the idea of partitioning clauses into variable indecomposable subclauses further. Based on the prefix stability property we will define a finer nested and tree-like partition of clauses, on the basis of which decidability will be proved by an inductive argument.

6.2 The matrix term representation of clauses

Assuming that m denotes the maximal length of any term in the input set $S \subseteq \mathbf{S}$, we encode any clause in $\mathbf{S}$ by a set of terms of length $m+1$. The encoding is a two step conversion. By introducing new blank symbols which we append to terms shorter than m we obtain terms with uniform lengths. We convert literals to terms by introducing additional new constant symbols for positive and negative occurrences of predicate symbols which we append to the terms. For example, if $m = 3$ the clause

$$P[\alpha\underline{\beta}] \vee \neg P[\alpha\gamma\underline{\delta}] \vee Q[\underline{\epsilon}\gamma'] \quad \text{is represented by} \quad \boxed{\begin{array}{l} \alpha\ \underline{\beta}\ \underline{b}\ \underline{p} \\ \alpha\ \gamma\ \underline{\delta}\ \underline{p}' \\ \underline{\epsilon}\ \gamma'\ \underline{b}\ \underline{q} \end{array}}\,. \tag{6.1}$$

$\underline{b}$ is a special blank constant, and $\underline{p}$, $\underline{p}'$ and $\underline{q}$ are new and different constant symbols associated with P, $\neg P$ and Q, respectively. They are called *predicate constants*. In the described way any clause C of $\mathbf{S}$ is converted to a set T of terms all with length $m + 1$. T can be viewed as a matrix when we write the terms below each other as in (6.1). This *matrix term representation* of a clause helps visualising variable and prefix partitions. Prefix stability is preserved by the encoding because we append the new predicate constant. Prepending the new constants would not preserve prefix stability.

Suppose K be the finite set of constant symbols in the term representation of the input set S. Suppose X is the union of a finite sequence $X_1, \ldots, X_m$ of pairwise disjoint non-empty sets of variables. Define $\mathcal{T}(X, K)$ to be the set of all terms that are built from X and K with the restriction that

$$s \in \mathcal{T}(X, K) \quad \text{iff} \quad s = [u_1 \ldots u_m u_{m+1}],$$

where $u_{m+1} \in K$ and for each $1 \leq i \leq m$,

$$u_i \in \begin{cases} X_i, & \text{when } u_i \text{ is a variable, and} \\ K, & \text{when } u_i \text{ is a constant.} \end{cases}$$

Each X_i contains the variables that occur in position i of any term in $\mathcal{T}(X, K)$. Thus, $\mathcal{T}(X, K)$ is the class of strings of length $m+1$ built from K and X that satisfy the property T1 of Theorem 5.1.1. This means, each variable in any subset T of $\mathcal{T}(X, K)$ occurs at one fixed position in any term, but T is not necessarily prefix stable. T can be viewed as a matrix with $m+1$ columns and as many rows as there are terms in T.

Lemma 6.2.1 Let C be any basic path clause with maximal term length m. There is a one-one correspondence between C and its matrix term representation $T \subseteq \mathcal{T}(X, K)$ such that C is prefix stable iff so is T.

Proof. Not difficult. □

Reformulated for the term representation our goal is to show there is a bound for the size of any condensed and variable indecomposable and prefix stable set $T \subseteq \mathcal{T}(X, K)$. (By definition, a set T of terms is *condensed* if there is no substitution σ such that $T\sigma \subsetneqq T$.) Or, reformulated for the matrix term representation our goal is to show there is a bound for the height of any non-variant matrix representing a condensed, variable indecomposable and prefix stable set $T \subseteq \mathcal{T}(X, K)$.

We define two operations for dividing and reassembling $T \subseteq \mathcal{T}(X, K)$. Let s be a term and define

$$T(s) = \{[u_1 \dots u_i] \,|\, [s u_1 \dots u_i] \in T\}.$$

$T(s)$ is the *set of suffixes* of s occurring in T and may be viewed as a sub-matrix of T. For example, if T is the matrix of (6.1) then

$$T([\alpha]) = \boxed{\begin{matrix} \underline{\beta}\, \underline{b}\, \underline{p} \\ \gamma\, \underline{\delta}\, \underline{p}' \end{matrix}}\,.$$

Forming sub-matrices in this way satisfies a kind of associativity: For any terms s and t occurring in T,

$$(T(s))(t) = T([st]).$$

For reassembly the following operation is used.

$$s\,T = \{[s u_1 \dots u_i] \,|\, [u_1 \dots u_i] \in T\}.$$

Then, $s\,T(s)$, more precisely $s\,(T(s))$, denotes the subset of all terms in T which have prefix s. For example, if T is the matrix of (6.1) then

$$[\alpha]\,T([\alpha]) = \boxed{\begin{matrix} \alpha\, \underline{\beta}\, \underline{b}\, \underline{p} \\ \alpha\, \gamma\, \underline{\delta}\, \underline{p}' \end{matrix}}\,.$$

Prepending terms is monotone, that is, $T \subseteq T'$ implies $s\,T \subseteq s\,T'$. Subsequently, we write $T(\alpha)$ and $\alpha\,T(\alpha)$ instead of $T([\alpha])$ and $[\alpha]\,T([\alpha])$.

6.3 Variable partitioning

Initially, in this section, we suppose T is a set of terms of basic path logic (not necessarily in matrix form). The *variable partition* of a set T of terms is the finest partition into subsets of terms that do not have any variables in common. A set of terms is *variable indecomposable* (or *split*) iff its variable partition is a singleton set.

Let θ_v be a binary relation on a set T of terms defined by:

$$s\theta_v t \quad \text{iff} \quad s \text{ and } t \text{ have a variable in common or } s = t.$$

In general, θ_v is not an equivalence relation, because transitivity may fail. For example, $[\alpha\beta]\theta_v[\beta\gamma]$ and $[\beta\gamma]\theta_v[\delta\gamma]$, but we do not have that $[\alpha\beta]\theta_v[\delta\gamma]$. However, θ_v defined over terms of basic path logic has the following properties.

Lemma 6.3.1 Let T be a set of terms in basic path logic. Let s and t be terms in T with s not ground. Then

(i) $s\theta_v t$ iff for some i, $s|_i = t|_i$ is a variable.

(ii) $s\theta_v t$ iff s and t have a common prefix $[u_1 \dots u_i]$ with u_i a variable.

(iii) θ_v is an equivalence relation.

If s is ground then

(iv) $s\theta_v t$ iff $s = t$.

Proof. Not difficult, exploiting prefix stability. □

We adopt the standard notation. If θ is an equivalence relation over a set T then T/θ denotes the *partition* of T. The elements in the partition are pairwise disjoint subsets $s/\theta = \{t \in T \mid s\theta t\}$ of T, and are called *blocks*.

Theorem 6.3.2 Let T be a set of terms in basic path logic. Then

(i) T/θ_v coincides with the variable partition of T.

(ii) Each non-ground block in T/θ_v is uniquely associated with a term $t = [u_1 \dots u_i]$ where u_i is a variable and t is the prefix of every term in the block.

(iii) Any ground block in T/θ_v is a singleton set.

Proof. (i) Since θ_v is an equivalence relation, T/θ_v is a partition of T. Evidently, no two different blocks in T/θ_v share a variable. Because the blocks cannot be partitioned further, $T/\theta_v = \{s/\theta_v \mid s \in T\}$ is the variable partition of T.

(ii) and (iii) are immediate by (ii) of the previous lemma. □

Properties (ii) and (iii) are important and imply all terms in any block of the variable partition have one common prefix:

Corollary 6.3.3 Let $T \subseteq \mathcal{T}(X, K)$ be prefix stable.

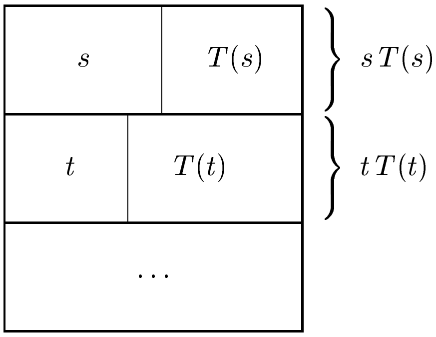


Figure 6.1: The variable partition T/θ_v

(i) Each block in the variable partition has the form $s\,T(s)$. Every non-ground block is of the form $[s\alpha]\,T([s\alpha])$, for some variable α.

(ii) If T is variable indecomposable then all terms in T have a common prefix and there is a prefix s in T such that $T = s\,T(s)$.

This proves that the variable partition of a set T may be viewed as a matrix arranged and divided as in Figure 6.1.

6.4 Prefix partitioning

For the definition of prefix partitions, it is essential that T is a set of terms of equal length and forms a matrix. Suppose $T \subseteq \mathcal{T}(X, K)$. We define a sequence of m equivalence relations θ_i (one for each of the first m columns in the matrix representation of T): For any $1 \le i \le m$, θ_i is the relation on T given by

$$s\theta_i t \quad \text{iff} \quad s \text{ and } t \text{ have a common prefix of length } i$$

We say a set T of terms is *prefix indecomposable* if the terms in T all have one common prefix (at least of length one).

For example, the θ_1-partition of the matrix in (6.1) consists of two blocks:

$$\boxed{\begin{array}{l} \alpha\ \underline{\beta}\ \underline{b}\ \underline{p} \\ \alpha\ \gamma\ \underline{\delta}\ \underline{p}' \end{array}} \quad \text{and} \quad \boxed{\underline{\epsilon}\ \gamma'\ \underline{b}\ \underline{q}}\ .$$

The common prefix of length one in the first block is $[\alpha]$ and the common prefix in the second block is $[\underline{\epsilon}]$. The θ_2- and θ_3-partitions are identical and consist of three blocks:

$$\boxed{\alpha\ \underline{\epsilon}\ \underline{b}\ \underline{p}}\ , \quad \boxed{\alpha\ \delta\ \underline{\gamma}\ \underline{p}'} \quad \text{and} \quad \boxed{\underline{\epsilon}\ \gamma'\ \underline{b}\ \underline{q}}\ .$$

Further examples of prefix partitions can be found in Figure 1.8 of the Preview.

Lemma 6.4.1 Let $T \subseteq \mathcal{T}(X, K)$.

(i) For every i, θ_i is an equivalence relation and partitions T.

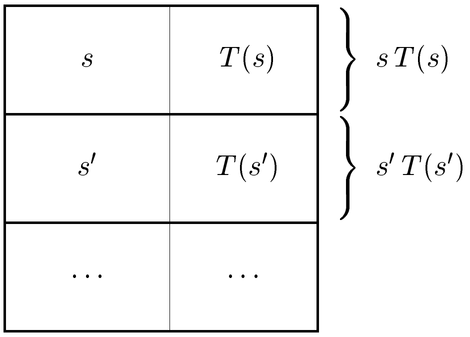


Figure 6.2: The prefix partition T/θ_i

(ii) The sequence $\theta_1, \dots, \theta_m$ is a descending chain, or equivalently, for any $s \in T$, $s/\theta_m \subseteq s/\theta_{m-1} \subseteq \dots \subseteq s/\theta_1$.

(iii) If T is prefix stable then, for any $s \in T$, $s/\theta_v \subseteq s/\theta_1$.

(iv) If T is prefix stable and variable indecomposable then T is prefix indecomposable.

Proof. (i) and (ii) are straightforward.

(iii) For every non-ground terms $s, t \in T$, $s\theta_v t$ means s and t share a variable. Prefix stability for variables implies $s\theta_1 t$, that is, s and t have a common prefix of at least length one. Otherwise, for ground terms $s\theta_v t$ iff $s = t$ (Lemma 6.3.1 (iv)) iff s and t have a common prefix of length one iff $s\theta_1 t$.

(iv) is by (iii), since T is prefix indecomposable iff $s/\theta_1 = T$, for any $s \in T$. □

Theorem 6.4.2 Let $T \subseteq \mathcal{T}(X, K)$ be prefix stable and let s be any term occurring in T. Then

(i) The blocks s/θ_i in T/θ_i are of the form $t\,T(t)$ for some t.

(ii) The blocks in $T(s)/\theta_i$ are of the form $t\,T([st])$ for some t.

Proof. Let t be the common prefix of length i of the terms in s/θ_i and s'/θ_i with s' some suffix of s. □

Therefore, like the blocks in the variable partition, the blocks in a prefix partition have the form $s\,T(s)$. Compare the prefix partition T/θ_i in Figure 6.2 with the variable partition T/θ_v in Figure 6.1.

The sequence, or nesting, of increasingly finer prefix partitions given by $\theta_1, \dots, \theta_m$ induces a tree-like structure on any prefix stable set of terms as depicted in Figure 6.3 (and Figure 1.9). The proof of (B) will be based on this structure. The next two theorems prove any sub-matrix $T(s)$ has the same properties as $s\,T(s)$.

The following is a corollary of Theorem 5.4.5, which says that prefix stability is preserved by taking subsets.[1]

[1] The next three results correct Theorem 4.3 of Schmidt (1997c).

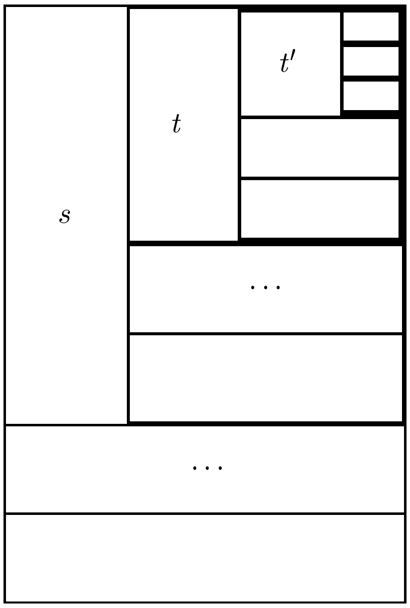


Figure 6.3: Nested partitioning of T

Corollary 6.4.3 Let $T \subseteq \mathcal{T}(X, K)$ be prefix stable. Then

(i) For every i, every block s/θ_i in T/θ_i is prefix stable.

(ii) For every s, $s\,T(s)$ and $T(s)$ are prefix stable.

In general, prefix stability is not preserved by prepending prefixes. However, a consequence of Theorem 5.4.6 is:

Corollary 6.4.4 Let $\{s\,T(s)\}_s$ be a collection such that $\{s\,T(s)\}_s \subseteq T/\theta_i$ and the $s\,T(s)$ are pairwise variable disjoint. If each $s\,T(s)$ is prefix stable then the union $\bigcup_s s\,T(s)$ is prefix stable.

For condensedness a stronger result holds.

Theorem 6.4.5 Let $T \subseteq \mathcal{T}(X, K)$ be prefix stable. The following statements are equivalent:

(i) T is condensed.

(ii) For every s, $s\,T(s)$ is condensed.

(iii) For every s, $T(s)$ is condensed.

Proof. (iii) implies (i), since $T = T([])$. We prove that if $T(s)$ is not condensed then $s\,T(s)$ is not condensed, which implies T is not condensed, that is, (i) implies (ii) implies (iii).

Suppose $T(s)$ is not condensed. Then there is a substitution σ such that $T(s)\sigma$ is a proper subset of $T(s)$. Since prepending a term s is order preserving we conclude $(s\,T(s))\sigma =$

$s(T(s)\sigma)$ is a proper subset of $s\,T(s)$ and this means $s\,T(s)$ is not condensed. Since $s\,T(s)$ and $T \setminus s\,T(s)$ have no common variables (by prefix stability) we derive that

$$T\sigma = (T \setminus s\,T(s))\sigma \cup s\,T(s)\sigma = (T \setminus s\,T(s)) \cup s\,T(s)\sigma.$$

This is a proper subset of $(T \setminus s\,T(s)) \cup s\,T(s) = T$. Hence T is not condensed. $\square$

6.5 Decidability results

We now show the process of repeatedly partitioning a matrix T and its sub-matrices $T(s)$ by common prefixes has a finitely bounded number of levels, that is, each partition consists of a bounded number of blocks/sub-matrices of bounded size/height.

The underlying inductive construction is this. Let T^i denote any matrix of width i, more precisely, any sub-matrix of suffixes of any $T \subseteq \mathcal{T}(X, K)$. Starting with the smallest matrices T^1 of width one, we will iteratively build the matrix T^{m+1}, by combining at each level a set of variable indecomposable and non-variant T^i with a common constant or variable u to form a matrix T^{i+1}. That is, $T^{i+1} = u\,T^{i+1}(u)$ for some u. By the results of the previous sections, the construction in such that each T^i is condensed and prefix stable.

Theorem 6.5.1 Let m be a given natural number. There is a bound for the size of any condensed set T of terms in the basic path logic with length at most m.

Proof. It suffices to prove any variable indecomposable and condensed T is finitely bounded, because, if this is the case then there are finitely many non-variant such T. The proof is by induction on the length i of suffixes in the matrix representation of T (Lemma 6.2.1) using the nested tree-structure induced by prefix partitioning (Figure 6.3).

The base case is easy. The last column of any T consists of just (predicate) constants. So, any variable indecomposable and condensed T^1 of width one is ground, and thus a singleton set. In other words, the height of any variable indecomposable and condensed T^1 is $n_0 = 1$.

The inductive hypothesis says, there is a bound n_i, say, for the height of any variable indecomposable and condensed T^i of width $i \leq m$. This implies there are finitely many such non-variant T^i, and consequently:

$('')$ There is a bound n_{i+1} for the height of any condensed T^i.

In the inductive step, we assume T^{i+1} of width $i+1$ is variable indecomposable and condensed. Corollary 6.3.3 implies $T^{i+1} = u\,T^{i+1}(u)$ for some constant or variable u. By Corollary 6.4.3 and Theorem 6.4.5, $T^{i+1}(u)$ is prefix stable and condensed. Hence, using $('')$, the height of $T^{i+1}(u)$ is bounded by n_{i+1}, which means T^{i+1} is, too.

This proves there is a bound n_{m+1} for the height (size) any variable indecomposable and condensed T^{m+1}. $\square$

With this theorem we have achieved our goal (as set out for this thesis) of exhibiting the existence of a size bound and hence a variable bound for basic path clauses in any saturated class **S**. This is the most important theorem of this chapter from which decidability for a range of path logics and the corresponding modal logics follows. As syntactic unification does not cause term lengths to increase (Theorem 5.3.3), it immediately follows that any procedure $\mathcal{R}_{\text{COND}}$ with empty theory terminates for any finite input set of path clauses. Formally:

Corollary 6.5.2 Any complete resolution procedure with condensing is a decision procedure for the satisfiability problem of a finite set of finite clauses of the basic path logic.

Proof. There is no growth of terms (Theorem 5.3.3), the basic path logic is closed under resolution with condensing (Theorem 5.4.9), by Theorem 6.5.1 there is no unbounded growth of the size of clauses, and any compatible refinement will merely reduce the size of **S**. □

Formulated more generally:

Theorem 6.5.3 Any complete resolution procedure with condensing and possibly a normalisation function N_E is a decision procedure for the satisfiability problem of a finite set of finite clauses in path logics, provided

(i) a term depth bound exists,

(ii) unification is decidable, and

(iii) the normalisation function is effective and returns basic path clauses.

Although any input set is a set of basic path clauses without any non-constant occurrences of functional terms, condition (iii) is important as theory unification may introduce non-constant functional terms.

Condition (i) can be interpreted in two ways. Namely, a term depth bound exists for the particular resolution procedure. Or, an a priori term depth bound exists for the particular path logic. The latter can be extracted from the literature on modal logic, for example, from proofs of the finite model property or termination proofs of tableaux procedures. The given term depth bounds can then be used in a simple blocking mechanism to stop the theorem prover from generating clauses of depth greater than the given value.

Theorem 6.5.3 defines exactly the class of path logics (in general, with a theory) for which unrestricted resolution plus condensing terminates always. Because the translation of modal formulae into path clauses has linear time and space complexity, it is immediate that Theorem 6.5.3 provides a decidability result for the satisfiability problem in those modal logics which can be embedded in path logics such that conditions (i)–(iii) hold. We do not know exactly which modal logics meet the conditions of Theorem 6.5.3. But, all three conditions are met by the following modal logics:

Corollary 6.5.4 Resolution and condensing combined with any compatible refinement strategies is a decision procedure

(i) for basic path logic and for the translation of *K*, *KD*, *S5*, and the multi-modal versions of *K* and *KD*,

(ii) for the translation of *KT*, and

(iii) for *KD4* and *S4*.

Proof. (i) follows by Theorem 5.1.2, Corollary 6.5.2 and the remarks on p. 84 about embedding *S5* into the basic path logic.

(ii) Use E-resolution, condensing and normalisation modulo E as defined in Chapter 5.

(iii) Use E-resolution and condensing with unification and normalisation defined as in Chapter 5, and an a priori term depth bound extracted from Ladner (1977), for example. □

Another important question is, are there practical elementary resolution decision procedures for transitive modal logics? For practical purposes the solution using an artificial term depth bound is poor (Hustadt, Schmidt and Weidenbach 1998). Bounds implicit in proofs of the finite model property or termination proofs of tableaux procedures are highly non-optimal. Consequently, the search space is unrealistically large even for small input formulae. For example, the maximal depth of the trees underlying the models constructed by Ladner (1977) for *S4* is n^2, where n is the number of symbols in the input formula. It is clear from inspecting the algorithm that a better bound can be given, see Viganò (1997), for example. Further discussion of the problem can be found in Section 7.5 of the next chapter.

6.6 Bounds for literals and variables

The potential search space for unrefined resolution is enormous. In this section we will derive an explicit estimation based on an alternative direct proof method of decidability (first published in Schmidt (1997c, 1997d)). This requires a technical encoding of variables that exploits the tree-like structure of the prefix partitions (Figure 6.3).

Our strategy is, to show there is an embedding of the variables of T into a finite Boolean algebra. It is natural to look for an embedding that encodes the variables of a given set T in terms of the finite entities of T, that is, in terms of the constants. The problem is, how do we deal with condensing. Ideally we want the encoding to have a natural property that captures the condensedness property of the clause. The encoding we use is weaker. It has a natural property that is necessary but not sufficient for condensedness. We encode variables by sets of tuples that represent the set of suffixes of the given variable and the necessary property for a T to be condensed is the *antichain property*. It requires that the encodings of variables with the same prefix form an antichain. (An *antichain* is a subset of pairwise incomparable elements in a poset $(P, \subseteq)$.) The encoding of variables will be by a family of embeddings $\{f_s\}_s$. Each f_s will map any variable α with prefix s to a tuple-encoding of the sub-matrix $T([s\alpha])$ that represents the set of suffixes of the variable. The most difficult part of the proof is showing that, if there is a subset relationship between the encodings of the pairs of variables (that is, the antichain property is not true) then one of the corresponding blocks is redundant with respect to condensing (Theorem 6.6.3).

Let $T \subseteq \mathcal{T}(X, K)$ be prefix stable. Define f_s recursively as follows: For every prefix $[su]$ in (any term of) T

(i) if s has length $m-1$ and $u \in X_m \cup K$ then

$$f_s(u) = \begin{cases} \{\underline{p} \mid [s u \underline{p}] \in T\} & \text{if } u \text{ is a variable} \\ u & \text{otherwise, and} \end{cases}$$

(ii) if s has length $0 \leq i < m-1$ and $u \in X_{i+1} \cup K$ then:

$$f_s(u) = \{(f_{[su]}(u_{i+2}), \ldots, f_{[suu_{i+2} \ldots u_{m-1}]}(u_m), \underline{p}) \mid [suu_{i+2} \ldots u_m \underline{p}] \in T\}$$

if u is a variable, and $f_s(u) = u$ otherwise.

For technical reasons the f_s are defined also for constants. The restrictions of each f_s to constants are identity mappings. Restricted to variables each f_s maps $u = \alpha$ to a set of tuples that encodes the set of suffixes of α in T. s in the index of f_s is the unique prefix of α in T.

The encoding is best explained with an example. Consider the matrix of (6.1):

$$\begin{array}{|l|}\hline \alpha\ \underline{\beta}\ \underline{b}\ \underline{p} \\ \hline \alpha\ \gamma\ \underline{\delta}\ \underline{p}' \\ \hline \underline{\epsilon}\ \gamma'\ \underline{b}\ \underline{q} \\ \hline \end{array} \qquad \begin{array}{l} \\ f_{[\alpha]}(\gamma) = \{(\underline{\delta}, \underline{p}')\} \\ f_{[\underline{\epsilon}]}(\gamma') = \{(\underline{b}, \underline{q})\} \end{array} \qquad \begin{array}{l} f_{[]}(\alpha) = \{(\underline{\beta}, \underline{b}, \underline{p}), \\ \qquad (\{(\underline{\delta}, \underline{p}')\}, \underline{\delta}, \underline{p}')\} \\ \\ \end{array}$$

The encodings of the variables α, γ and γ' are defined inductively starting with the variables closest to the end of any term. The suffix of γ is $[\underline{\delta}\underline{p}']$ and its encoding by $f_{[\alpha]}$ (because $[\alpha]$ is the prefix of γ) is the singleton set containing the tuple $(\underline{\delta}, \underline{p}')$. Similarly, the encoding of the suffix $[\underline{b}\,\underline{q}]$ of γ' by $f_{[\underline{\epsilon}]}$ is the tuple $(\underline{b}, \underline{q})$. The variable α occurs twice and the encoding of its two suffixes by $f_{[]}$ is $(\underline{\beta}, \underline{b}, \underline{p})$ and $(f_{[\alpha]}(\gamma), \underline{\delta}, \underline{p}') = (\{(\underline{\delta}, \underline{p}')\}, \underline{\delta}, \underline{p}')$.

Denote the domain of each f_s restricted to variables by X_s. Any X_s is the set of variables of T that all have prefix s and prefix stability ensures that distinct X_s and $X_{s'}$ are disjoint. In the base case (i) of the definition of f_s, the restriction of f_s to variables is a mapping from X_s to a subset of the set of predicate constants. For example

$$f_s(\beta) = \{\underline{p}, \underline{q}\} \quad \text{for} \quad \begin{array}{|l|}\hline s\ \beta\ \underline{p} \\ s\ \beta\ \underline{q} \\ \hline \end{array}\,.$$

If s has length $m-2$ as in

$$\begin{array}{|l|}\hline s\ \alpha\ \beta\ \underline{p} \\ s\ \alpha\ \beta\ \underline{q} \\ s\ \alpha\ \underline{\gamma}\ \underline{q} \\ \hline \end{array}$$

the restriction of f_s to variables is a mapping from X_s to a subset of the product $(K \cup \mathbf{2}^K) \times K$. In this example, $f_s(\alpha) = \{(\{\underline{p}, \underline{q}\}, \underline{p}),\ (\{\underline{p}, \underline{q}\}, \underline{q}),\ (\underline{\gamma}, \underline{q})\}$. In general, the restriction of f_s to variables is a mapping $X_s \longrightarrow \mathbf{2}^{B_{i+1}}$ where s is a prefix in T of length $0 \leq i < m$ and B_{i+1} is defined recursively by

$$\begin{aligned} B_m &= K \\ B_i &= (K \cup \mathbf{2}^{B_{i+1}}) \times \ldots \times (K \cup \mathbf{2}^{B_m}) \times K \quad \text{for any } 1 \leq i < m. \end{aligned}$$

Since K is finite, each $\mathbf{2}^{B_{i+1}}$ is a finite Boolean algebra. Thus:

> *Provided the restriction of f_s to variables is an injection of X_s into $\mathbf{2}^{B_{i+1}}$*, it follows that (i) the size of each $f_s(\alpha)$ is bounded by $\mathrm{card}(B_{i+1})$ for i the length of s, and (ii) each X_s is bounded by $2^{\mathrm{card}(B_{i+1})}$.

X_s is the set of variable with prefix s that occur at position $i+1$, meaning $X_s \subseteq X_{i+1}$. So, $X_{i+1} = \bigcup_{s \in I_i} X_s$ where I_i is the set of prefixes s of length i in T. It then follows there are also bounds for the sizes of any X_{i+1} that are used to build any prefix stable and condensed set $T \subseteq \mathcal{T}(X, K)$ with given depth m. This provides then another proof for the existence of variable and literal bounds (Theorem 6.5.1).

Next we prove the restriction of f_s to variables is indeed an injection. f_s is not an injection for every T, especially not if T is not condensed, because both variables α_1 and α_2 in

$$\boxed{\begin{array}{l} s\,\alpha_1\,\underline{p} \\ s\,\alpha_2\,\underline{p} \end{array}}$$

for example, are mapped to $\{\underline{p}\}$.

We need the following mapping. Let α be an arbitrary variable in $X_s \subseteq X_{i+1}$ with prefix s. For any T (not necessarily condensed) define a mapping

$$h_\alpha : \alpha\,T([s\alpha]) \longrightarrow f_s(\alpha)$$

from the sub-matrix $\alpha\,T([s\alpha])$ of suffixes with prefix α to the encoding of α specified by

$$h_\alpha([\alpha u_{i+2}\dots u_m\underline{p}]) = (f_{[s\alpha]}(u_{i+2}),\dots,f_{[s\alpha u_{i+2}\dots u_{m-1}]}(u_m),\underline{p}).$$

That is, h_α maps any suffix $[\alpha u_{i+2}\dots u_m\underline{p}]$ beginning with α to the tuple of encodings of the u_{i+j} following α.

Lemma 6.6.1 Let $T \subseteq \mathcal{T}(X,K)$ be prefix stable. Let α be a variable in X_{i+1} with prefix s. Then h_α is a surjective function.

Proof. We readily see that h is well-defined, total and single-valued. That h is onto, that is, for any tuple $\overline{f}$ in $f_s(\alpha)$ a term t exists in $\alpha\,T([s\alpha])$ such that $h_\alpha(t) = \overline{f}$, follows from the definition of f_s. □

We will show that h_α is a bijection, provided T (or $\alpha\,T([s\alpha])$) is condensed (Theorem 6.6.6).

Lemma 6.6.2 Let $T \subseteq \mathcal{T}(X,K)$ be prefix stable. Let α and β be two different variables in T with a common prefix s. If $f_s(\alpha) = f_s(\beta)$ then

(i) T is not condensed, and

(ii) there is a substitution σ such that $(\alpha\,T([s\alpha]))\sigma \subseteq \beta\,T([s\beta])$, provided $\mathrm{card}(\alpha\,T([s\alpha])) \leq \mathrm{card}(\beta\,T([s\beta]))$.

Proof. (ii) implies (i): Suppose $\alpha\,T([s\alpha])$ is smaller in size than $\beta\,T([s\beta])$. We will construct a substitution σ from $f_s(\alpha)$ and $f_s(\beta)$ such that $(\alpha\,T([s\alpha]))\sigma$ is a subset of $\beta\,T([s\beta])$. It follows that $s\,T(s)\sigma$ is a proper subset of $s\,T(s)$. Therefore, $s\,T(s)$ is not condensed and by Theorem 6.4.5, T cannot be condensed either.

(ii) Assume s has length i. Consider any suffix of α that has the form

$$t_1 = [t\gamma u_{i+j+2}\dots u_m\underline{p}]$$

with t being a string of constants $[\underline{\alpha}_1\dots\underline{\alpha}_j]$. Its tuple encoding is

$$\overline{f} = (\underline{\alpha}_1\dots\underline{\alpha}_j, f_{[s\alpha t]}(\gamma), \widehat{u_{i+j+2}},\dots,\widehat{u_m},\underline{p}),$$

where $\widehat{u} = f_{s'}(u)$ for any u with prefix s'. The tuple $\overline{f}$ belongs to both $f_s(\alpha)$ and $f_s(\beta)$. Since h_β is onto (Lemma 6.6.1), there is a term t_2 in $\beta\,T([s\beta])$ such that $h_\beta(t_2) = \overline{f}$. This

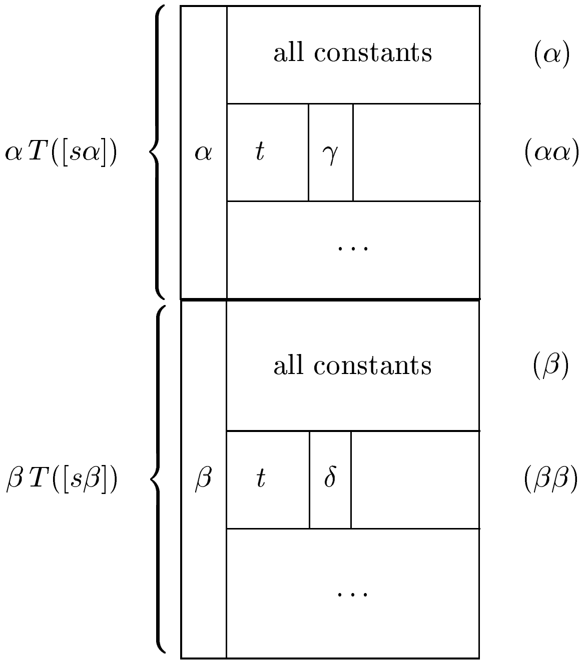


Figure 6.4: $\alpha\, T([s\alpha]) \cup \beta\, T([s\beta])$

means, there is a variable δ in $\beta\, T([s\beta])$ with prefix $[s\beta t]$ such that $f_{[s\alpha t]}(\gamma) = f_{[s\beta t]}(\delta)$. That is, in $f_s(\beta)$ the tuple $\overline{f}$ is associated with some suffix

$$t_2 = [t\delta v_{i+j+2} \dots v_m \underline{p}]$$

of β. t_1 and t_2 have a similar form: They are equal at all positions filled with constants and their variables form pairs determined by equal encodings. t_1 and t_2 are variable disjoint (otherwise prefix stability is violated). Hence, there is a most general substitution σ' such that $t_1\sigma' = t_2$ with σ' determined by equal $\widehat{u}$ and $\widehat{v}$ stemming from u and v that are variables. For example, the binding $\gamma \mapsto \delta$ is in σ'. Let θ' be the union of all such most general unifiers θ with $t_1\theta \in T([s\beta])$ for $t_1 \in T([s\alpha])$. Then $\sigma = \{\alpha \mapsto \beta\} \cup \theta'$ is a substitution such that $(\alpha\, T([s\alpha]))\sigma \subseteq \beta\, T([s\beta])$. □

The next lemma is slightly more general and relates condensedness to antichains of the variable encodings.

Lemma 6.6.3 Let $T \subseteq \mathcal{T}(X, K)$ be prefix stable. Let α and β be two different variables in T with a common prefix s. If $f_s(\alpha) \subseteq f_s(\beta)$ then there is a non-trivial substitution σ such that $(\alpha\, T([s\alpha]))\sigma \subseteq \beta\, T([s\beta])$.

Proof. Suppose $\alpha \neq \beta$ and $f_s(\alpha) \subseteq f_s(\beta)$. Suppose s has length i. As in the previous proof our aim is to construct a suitable substitution σ based on the information provided by the relation $f_s(\alpha) \subseteq f_s(\beta)$. Again, σ will include the binding $\sigma_0 = \{\alpha \mapsto \beta\}$ plus other substitutions θ.

Given that $f_s(\alpha) \subseteq f_s(\beta)$, the sets $\alpha\, T([s\alpha])$ and $\beta\, T([s\beta])$ can be viewed as depicted in Figure 6.4. $T([s\alpha])$ is the union of the set

$$(\alpha) \qquad \{[\ldots \underline{p}] \mid [s\alpha \ldots \underline{p}] \in T \text{ with the `\ldots' being constants only}\}$$

and sets of the form

$$(\alpha\alpha) \qquad [t\gamma]\, T([s\alpha t\gamma])$$

for γ any variable following α so that at the positions between α and γ only constants occur (that is, t is a string of constants). Analogously, $T([s\beta])$ is the union of a (β)-part of constants only, and $(\beta\beta)$-parts of the form $[t'\delta]\, T([s\beta t'\delta])$. Because $f_s(\alpha) \subseteq f_s(\beta)$, the (α)-part of $T([s\alpha])$ is a subset of the (β)-part of $T([s\beta])$.

Let

$$\overline{f} = (\underline{\alpha}_1 \ldots \underline{\alpha}_j, f_{[s\alpha t]}(\gamma), \widehat{u_{i+j+2}}, \ldots, \widehat{u_m}, \underline{p}),$$

be an arbitrary element of $f_s(\alpha)$. It is also an element of $f_s(\beta)$. Then, t and t' coincide. By the argument also used in the previous proof we know a variable δ exists in $T([s\beta])$ such that $f_{[s\alpha t]}(\gamma) = f_{[s\beta t]}(\delta)$. Applying the previous lemma, we infer that there is a substitution θ such that

$$(*) \qquad (\gamma\, T([s\alpha t\gamma]))\theta \subseteq \delta\, T([s\beta t\delta]).$$

θ contains the binding $\gamma \mapsto \delta$.

Therefore, a substitution σ that is such that $(\alpha\, T([s\alpha]))\sigma \subseteq \beta\, T([s\beta])$ as required by the theorem is

$$\sigma_0 \cup \{\theta \mid \theta \text{ satisfies } (*) \text{ with } \gamma \text{ a variable in some } (\alpha\alpha)\text{-part of } \alpha\, T([s\alpha])\}.$$

This completes the proof. □

If there is substitution σ such that $(\alpha\, T([s\alpha]))\sigma \subseteq \beta\, T([s\beta])$ and α and β are different then the block $[s\alpha]\, T([s\alpha])$ of T is redundant, and both $s\, T(s)$ and T are not condensed. For condensed T and $s\, T(s)$ a necessary requirement is then that $f_s(\alpha) \not\subseteq f_s(\beta)$, for every pair of different variables. In other words:

Theorem 6.6.4 Let $T \subseteq \mathcal{T}(X, K)$ be prefix stable. If T is condensed then f_s satisfies the *antichain property*: The set of images of X_s by f_s forms an antichain.

Proof. By Lemma 6.6.3. □

The antichain property of the f_s is a necessary but not a sufficient condition for condensedness. One problem are constants. The set

$$\boxed{\begin{array}{c} s\ \alpha\ \underline{\gamma}\ \underline{p} \\ s\ \beta\ \delta\ \underline{p} \end{array}}$$

is not condensed but it does satisfy the antichain property, because $f_s(\alpha) = \{(\underline{\gamma}, \underline{p})\}$ and $f_s(\beta) = \{(\{\underline{p}\}, \underline{p})\}$ are not comparable. The redundancy eliminating substitution is $\{\delta \mapsto \underline{\gamma}\}$

and involves binding a constant which is not conveyed by the encoding. Here is another example:

$s\,\gamma_1\,\beta_1\,\underline{\delta}\,\underline{p}$
$s\,\gamma_1\,\beta_1\,\alpha_1\,\underline{q}$
$s\,\gamma_2\,\beta_2\,\alpha_2\,\underline{q}$

$f_s(\gamma_1)$ and $f_s(\gamma_2)$ are not comparable, but the set is not condensed.

Theorem 6.6.5 Let $T \subseteq \mathcal{T}(X, K)$ be prefix stable and condensed. For any prefix s occurring in T, the restriction of f_s to variables is an injection from X_s into $\mathbf{2}^{B_{i+1}}$.

Proof. Suppose X_s non-empty. f_s is total and it is single-valued (that is, if $\alpha = \beta$ then $f_s(\alpha) = f_s(\beta)$), hence, it is a function. By Lemma 6.6.2 (i), f_s is one-one. □

Now it follows that for condensed T, h_α is an injection, and more precisely:

Theorem 6.6.6 Let $T \subseteq \mathcal{T}(X, K)$ be prefix stable. Let α be any variable in X_{i+1} with prefix s. If T is condensed then h_α is a bijection.

Proof. In Lemma 6.6.1 we showed that h_α with α a variable in T (not necessarily condensed) is a surjection. Let

$$t_1 = [\alpha u_{i+2} \ldots u_m \underline{p}] \quad \text{and} \quad t_2 = [\alpha v_{i+2} \ldots v_m \underline{q}]$$

be arbitrary terms in $\alpha\, T([s\alpha])$ and assume $h_\alpha(t_1) = h_\alpha(t_2)$. This means $f_{[s\alpha]}(u_{i+2}) = f_{[s\alpha]}(v_{i+2})$, $f_{[s\alpha u_{i+2}]}(u_{i+3}) = f_{[s\alpha v_{i+2}]}(v_{i+3})$, etcetera. For every constant $\underline{\beta}$ occurring in $\alpha\, T([s\alpha])$, $f_s(\underline{\beta}) = \underline{\beta}$. This means t_1 and t_2 are equal at all positions that are filled with constants, in particular, $\underline{p} = \underline{q}$. Thus, if t_1 contains no variable, neither does t_2 and $t_1 = t_2$.

If t_1 does contain a variable, let γ be the first variable following α, that is, let t is a string of constants and

$$t_1 = [\alpha t \gamma u_{i+j+1} \ldots u_m \underline{p}], \quad \text{then} \quad t_2 = [\alpha t \delta v_{i+j+1} \ldots v_m \underline{p}]$$

for some suitable $v_{i+j+1}, \ldots, v_m$. Since $h_\alpha(t_1) = h_\alpha(t_2)$, $f_{[\alpha t]}(\gamma) = f_{[\alpha t]}(\delta)$ and this implies that $\gamma = \delta$, using Theorem 6.6.5. Repeat the argument for the variables following γ.

We conclude that if $h_\alpha(t_1) = h_\alpha(t_2)$ then $t_1 = t_2$, that is, h_α is one-one. □

This proves there is a one-one correspondence between $\alpha\, T([s\alpha])$ and $f_s(\alpha)$, provided T is condensed. Or, if T is not condensed then there is a one-one correspondence between the condensation of $\alpha\, T([s\alpha])$ and $f_s(\alpha)$. Thus, $\alpha\, T([s\alpha])$ and $f_s(\alpha)$ have the same cardinality.

Recall (from the beginning of the section), f_s maps the variables in X_s to $\mathbf{2}^{B_{i+1}}$, where $B_m = K$ and $B_i = (K \cup \mathbf{2}^{B_{i+1}}) \times B_{i+1}$ for any $1 \leq i < m$. We let k denote the cardinality of K and b_i the cardinality of the sets B_i. The b_i are given by

$$b_m = k \quad \text{and} \quad b_i = (k + 2^{b_{i+1}}) \cdot b_{i+1} \quad \text{for } 0 \leq i < m.$$

Lemma 6.6.7 Let $T \subseteq \mathcal{T}(X, K)$ be prefix stable and condensed. Let K be a finite set of constants. Then:

(i) For any variable α, $\mathrm{card}(f_s(\alpha)) = \mathrm{card}(\alpha\, T([s\alpha])) \leq b_{i+1}$, when s has length i.

(ii) The size of any non-ground block $[s\alpha]\, T([s\alpha])$ in the variable partition of T is bounded by b_{i+1}, when s has length i.

Let s be any given prefix of length i. Then:

(iii) $\mathrm{card}(X_s) \leq 2^{b_{i+1}}$.

(iv) There are at most $2^{b_{i+1}}$ many non-variant and condensed sub-matrices of the form $[s\alpha]\, T([s\alpha])$.

Proof. Since $f_s : X_s \longrightarrow \mathbf{2}^{B_{i+1}}$ is an injection (Theorem 6.6.5) it follows $\mathrm{card}(f_s(\alpha)) \leq b_{i+1}$ and $\mathrm{card}(X_s) \leq 2^{b_{i+1}}$. This proves (i) and (iii). (iv) follows from (iii), and (ii) is by (i) and Corollary 6.3.3 (i). □

An instance of (ii) gives a bound for the cardinality of any variable indecomposable and condensed non-ground set T, namely b_1. The bound is an m-story exponential function and it is highly non-optimal, because we know (from Theorem 6.6.4), f_s maps the variables in X_s not onto $\mathbf{2}^{B_{i+1}}$ but onto antichains in $\mathbf{2}^{B_{i+1}}$.

A result by Sperner (1928) gives a bound for the maximal cardinality of any antichain in a finite power set algebra. Some more notation is needed. Let $[n]$ denote the set $\{1, \ldots, n\}$ of natural numbers. The family of subsets of $[n]$ is denoted by $\mathbf{2}^{[n]}$, and the family of k-element subsets by $\binom{[n]}{k}$. $\lfloor x \rfloor$ denotes the largest integer which is not greater than a given real number x.

Theorem 6.6.8 (Sperner 1928) Let $n \geq 2$ be an integer. Let $A \subseteq \mathbf{2}^{[n]}$ satisfying $X \not\subseteq Y$ for all $X, Y \in A$ with $X \neq Y$.

(i) Then $\mathrm{card}(A) \leq \binom{n}{\lfloor n/2 \rfloor}$.

(ii) This bound is best possible and it is attained iff

$$A = \begin{cases} \binom{[n]}{n/2} & \text{if } n \text{ is even} \\ \binom{[n]}{(n-1)/2} \quad \text{or} \quad \binom{[n]}{(n+1)/2} & \text{if } n \text{ is odd.} \end{cases}$$

The following two functions define improved bounds for the number of variables in the first position of a sub-matrix T^i of width i and the cardinality of T^i, for T^i condensed. Let

$$a_0 = 0 \quad \text{and} \quad a_i = \binom{n_{i-1}}{\lfloor n_{i-1}/2 \rfloor} \qquad \text{for } 0 < i \leq m$$

and

$$n_0 = k \quad \text{and} \quad n_i = (k + a_i) \cdot n_{i-1} \qquad \text{for } 0 < i \leq m.$$

Theorem 6.6.9 The cardinality of any condensed and prefix stable $T \subseteq \mathcal{T}(X, K)$ is bounded by n_m.

Proof. By induction on the width i of the sub-matrices T^i of suffixes determined by nested prefix partitioning (Figure 6.3) embodied in Corollaries 6.4.3 and 6.4.4 and Theorem 6.4.5.

In the base case consider any condensed set T^1 of terms with length one. T^1 is a set of predicate constants and a proper subset of K. The size of any T^1 is less than k. Thus, $a_0 = 0$ and $n_0 = k$.

Assume the number of variables in the first position of any condensed and prefix stable T^i is at most a_{i-1} and $\mathrm{card}(T^i) \leq n_{i-1}$. Any T^{i+1} is the union of non-variant matrices of the form $\alpha T^i(\alpha)$ and of the form $\beta T^i(\beta)$, with $T^i(\alpha)$ and $T^i(\beta)$ condensed and prefix stable. There are at most as many non-variant $\alpha T^i(\alpha)$ as there are antichains in $\mathbf{2}^{B_{m-i}}$ (Theorem 6.6.4). By Sperner's theorem and the inductive hypothesis there are at most $\binom{n_{i-1}}{\lfloor n_{i-1}/2 \rfloor} \cdot n_{i-1}$ of those. Thus $\mathrm{card}(T^{i+1}) \leq a_i \cdot n_{i-1} + k \cdot n_{i-1} = n_i$.

The original T has width $m+1$ and the required bound for T is then n_m. □

The bound n_m is a nested product of nested binomials and is an improvement over the bound b_1 which is a product of nested exponential functions. However, the new bound remains an m-story exponential, because:

Lemma 6.6.10 For any $0 < i \leq m$, there is some $c > 0$ such that $a_i > 2^{c \cdot n_{i-1}}$.

Proof. It suffices to prove that for any positive integer n, there is a value $c > 0$ such that $\binom{n}{\lfloor n/2 \rfloor} > 2^{c \cdot n}$. We consider only the case that n is a natural number divisible by four. Then both n and $n/2$ are even.

$$\binom{n}{\lfloor n/2 \rfloor} = \binom{n}{n/2} = \frac{n \cdot (n-1) \cdot (n-2) \cdot \ldots \cdot (n/2+2) \cdot (n/2+1)}{n/2 \cdot (n/2-1) \cdot \ldots \cdot 3 \cdot 2}$$
$$= \frac{2 \cdot (n-1) \cdot 2 \cdot \ldots \cdot 2 \cdot (n/2+1)}{n/4 \cdot (n/4-1) \cdot \ldots \cdot 3 \cdot 2} > 2^{n/4}$$

since every alternate value in the numerator is an even value divisible by some value of first $n/4$ values in the denominator. Thus, c for n divisible by four is $1/4$. Analogously, we can verify that the claim is true for all positive n. □

It now follows easily that a_i and n_i for $0 < i \leq m$ are exponential functions in which the level of exponentiation increases with i.

Let $\mathbf{S}$ be any class of *variable indecomposable, condensed and non-variant clauses* built from a given finite number of predicate and constant symbols with maximal path length m.

Theorem 6.6.11 Let k be the number of constant symbols in the matrix term representation of $\mathbf{S}$.

(i) The number of literals in any variable indecomposable clause of $\mathbf{S}$ is at most n_{m-1}.

(ii) The size of $\mathbf{S}$ is at most n_m.

Proof. (i) The largest possible variable indecomposable and condensed set in matrix form that is non-ground has the form $\alpha T^m(\alpha)$. It is bounded in size by n_{m-1} using the previous theorem (or b_1 using Lemma 6.6.7). All variable indecomposable and condensed ground clauses are singleton sets.

(ii) Each clause in $\mathbf{S}$ is variable indecomposable and condensed and translates to a variable indecomposable block of the form $[s\alpha]\, T([s\alpha])$ using the matrix term representation. The entire class translates to a prefix stable and condensed set $T \subseteq \mathcal{T}(X, K)$. Hence, (ii) follows by the previous theorem. □

6.7 Conclusion

The most important result is the decidability result for a range of path logics (Theorem 6.5.3). In conclusion we discuss its applications and significance. The result is important for several reasons. One, as we explained in Section 6.5, it renders resolution based decision procedures for a number of modal logics, including K, KD, KT, $KD4$, $S4$ and $S5$.

Two, the theorem can be applied in all areas of computer science in which modal logics are used. We already mentioned description logics used knowledge representation are in essence multi-modal logics. A reformulation of Corollary 6.5.4 (i) is: Resolution and condensing provides a decision procedure for the consistency problem of sentences in $\mathcal{ALC}$.[2] Resolution and condensing may also be used for doing arithmetic inside the algebraic counterparts of modal logics. The algebraic versions of K and $K_{(m)}$ are Boolean algebras with unary operators (Jónsson and Tarski 1951 & 1952). In particular, the Boolean set algebras with operators, called complex algebras, provide models for normal modal logics. Decidability for $K_{(m)}$ transfers to: Resolution and condensing provides a decision procedure for the satisfiability problem of algebraic identities of Boolean algebras with unary operators, as well as closure algebras, when assuming an artificial term depth bound.

We obtain a resolution decision result for basic non-optimised path logic and Herzig's (1990) class of ordered first-order formulae (which we briefly considered in Section 2.5). Because the result which allows us to permute quantifiers carries over (Corollary 3.2.1), satisfiability of any ordered formula can be decided by testing the satisfiability of the appropriate 'optimisation' of the ordered formula using resolution combined with condensing.

The decidability result for $S5$ has an interesting consequence for the monadic class. The *monadic class* is the class of first-order formulae without function symbols built from unary predicate symbols, only, using arbitrary quantification. It is commonly known that any monadic first-order formula ψ can be transformed into a formula of $S5$ by first transforming ψ into an equivalent formula in one variable and then using the inverse mapping of * given by:

$$\begin{aligned}
p^* &= P(x) \quad \text{for } p \text{ a propositional symbol} \\
\bot^* &= \bot \\
(\varphi \to \psi)^* &= \varphi^* \to \psi^* \\
(\Diamond\varphi)^* &= \exists x\ \varphi^*.
\end{aligned}$$

Since $S5$ is determined by the class of universal frames, that is, frames of the form (W, W^2) with truth is defined by

$$(W, W^2, \iota), x \models \Diamond\varphi \quad \text{iff} \quad (W, W^2, \iota), y \models \varphi \quad \text{for some } y \in W,$$

it immediately follows: φ is a theorem of $S5$ iff φ^* is a theorem of the monadic class. Joyner Jr. (1976) and others show that ordered resolution can be equipped with simplification techniques to ensure termination for the monadic class.[3] Corollary 6.5.4 (i) implies that

[2] Decidability for $\mathcal{ALC}$ is known (Schmidt-Schauß and Smolka 1991), who show that the coherence test (that is, the consistency test) and the subsumption problem (that is, the theoremhood problem) are decidable and moreover, both problems are PSPACE-complete and can be decided in linear space.

[3] Bachmair, Ganzinger and Waldmann (1993) present a decision procedure ordered resolution and superposition for the monadic class with equality.

satisfiability for any formula in the monadic class can be decided by restricted or unrestricted resolution and condensing on the optimised translations of the *S5*-reformulations. Because the translation cannot be effective, it is doubtful whether this approach is practical. (Testing satisfiability in *S5* is NP-complete whereas testing satisfiability in the monadic class is NEXPTIME-complete.)

The third and very practical consequence of our decidability result (especially, Theorem 6.5.2 for basic path logic) is that, *any* complete and fair implementation of a resolution theorem prover is a suitable (terminating) inference machine for modal reasoning, subsumption testing in description logics, arithmetic with algebraic identities and reasoning with ordered formulae. Even though condensing is not explicitly present in all resolution theorem provers, this is not a problem. Most theorem provers have factoring and subsumption deletion rules, which implies condensing is in fact implicit in any fair implementation. Because for many modal systems no special refinement strategies of resolution are required, all we have to implement, is a front end for transforming modal formulae to their path encodings and then we can use any resolution theorem prover at hand. As our results admit *any* compatible refinements, we have complete freedom to fine tune our theorem prover for specific purposes, for example, for improved performance or for integration into a resolution procedure predetermined by other considerations.

The potential search space for unrefined resolution is enormous. As established in Lemma 6.6.7 and Theorem 6.6.11, the worst case limits for the size of resolvents (and their number) are m-story exponential functions, where m is the maximal length of any term in the input set. It is not clear whether these bounds can be improved, but this issue is not of practical importance. A marked increase in efficiency can be achieved using the refinement techniques discussed in the next chapter.

Chapter 7

An ordered refinement

This chapter introduces a refined resolution calculus for the path logics of Chapter 5. In particular, we assume path logics with an equational theory with normal forms which can be described in the language of the basic path logic, namely for the combinations of D, T and 4. The refinement we study is ordered E-resolution on definitional clausal forms obtained by the technique of renaming. Both renaming and ordered resolution mean a considerable efficiency gain. Renaming is a very efficient method of producing clause sets, and the refinement of resolution by an ordering restriction will restrain the growth of terms noticeably. The ordered theory resolution procedure we will present combines the saturation approach of Bachmair and Ganzinger (1994, 1997) and E-resolution with normalisation of Plotkin (1972).

Section 7.1 presents a general definition of ordered E-resolution procedure with normalisation and proves completeness. Section 7.2 introduces an ordering which is compatible with the theories we concentrate on. Section 7.3 is devoted to the conversion of definitional forms by renaming. For path logics the resulting clauses are in a form, called prefix ordered, which are introduced and studied in Section 7.4. Though decidability for the path logics with term depth limit by ordered resolution follows from the main theorem of the previous chapter we give an independent and very simple argument for the basic path logic. Section 7.5 discusses decidability by resolution for $K4$ and $S4$.

7.1 Ordered E-resolution

In the literature we find two papers that deal with or mention ordered E-resolution. Baumgartner (1992) presents a general ordered theory resolution calculus and proves completeness by the excess literal parameter method (Anderson and Bledsoe 1970). And, Bachmair and Ganzinger (1997) mention (in a paragraph) how ordered E-resolution on the ground level can be treated in their framework of saturation up to redundancy. In this section we present the latter approach in detail, including a lifting lemma that renders a complete calculus also for non-ground inference. The calculus we consider is not specific to path logics.

As usual, completeness of ordered E-resolution will be achieved by proving completeness on the ground level and lifting it to the non-ground level. Proving ground completeness requires an enumeration of the set of all ground literals over the Herbrand universe.[1] The

[1]For some E an enumeration of a subset of all ground literals over the Herbrand universe is sufficient, for example, of the subset of unique normal forms.

Bachmair-Ganzinger method of proof requires

(i) any total well-founded strict ordering $\prec$ on the set of ground literals

with the two additional properties that for any atoms A and B,

(ii) $A \prec \neg A$ and

(iii) if $A \prec B$ then $\neg A \prec B$.

We refer to an ordering that has these three properties as *admissible*. An ordering on ground clauses is admissible if it is the multi-set extension of an admissible ordering on literals. An ordering of non-ground expressions is admissible if it is the extension of an admissible ordering of ground expressions as stipulated by: $A \prec B$ iff for any ground substitution σ, $A\sigma \prec B\sigma$. This implies that $\prec$ is stable under the application of substitutions: for an arbitrary pair A and B of non-ground atoms, if $A \prec B$ then $A\sigma \prec B\sigma$ for any substitution σ.

Commonly, the ordering $\prec$ arises from a *precedence*, which is a well-founded and total strict ordering on the function, predicate and logical symbols. The precedence is extended to a well-founded total strict ordering on multi-sets of ground terms, multi-sets of ground atom, multi-sets of ground literals and sets of ground clauses. These orderings, in turn, are extended to non-ground terms, literals, clauses and sets of clauses in such a way that well-foundedness, strictness and admissibility are preserved. There are a number of standard ways that the orderings can be defined from a given precedence. (The reader not familiar with these can consult Appendix B.1, or the survey paper of Dershowitz and Jouannaud (1990), or Bachmair and Ganzinger (1994, 1997).)

A sufficient condition enabling us to adopt the general Bachmair-Ganzinger method for proving completeness is: Compatibility of the ordering $\prec$ with E. It requires a total ordering on ground E-congruence classes, which we denote by $\prec^E$. By $\mathcal{A}$ we denote the set of all ground atoms in the Herbrand universe. $(\mathcal{A}, \prec)$ is a total well-founded and strict ordering. For any X and Y in $\mathcal{A}/{=_E}$ (or more generally $\mathbf{2}^{\mathcal{A}}$) define:

$$X \prec^E Y \quad \text{iff} \quad \forall A \in X \; \forall B \in Y \;\; A \prec B. \tag{7.1}$$

Any ordering $\prec$, for which $\prec^E$ orders $\mathcal{A}/{=_E}$ totally, is said to be *compatible with E*. It is not difficult to verify that when $\prec$ is stable under substitution then $\prec^E$ is, too.

Let $\prec$ be an admissible ordering on clauses which is compatible with E. Now, we define *ordered E-resolution with respect to $\prec$*. The ground resolution rule is:

Ground ordered E-resolution $\quad \dfrac{C \vee A_1 \vee \ldots \vee A_n \qquad C' \vee \neg A}{C \vee C'}$

provided (i) $A_1 =_E \ldots =_E A_n =_E A$, (ii) no atom in C is E-equivalent to any A_i, (iii) every atom in C is strictly smaller than any A_i with respect to $\prec$, and (iv) $\neg A$ is $\prec$-maximal in $C' \vee \neg A$.

(The first three conditions can be combined in two conditions: (i) the $A_1, \ldots, A_n, A$ all belong to the same congruence class $A/{=_E}$, (ii) every atom in C belongs to a congruence class strictly smaller than $A/{=_E}$ with respect to $\prec^E$.) The general resolution rule is:

Ordered E-resolution $$\frac{C \vee A_1 \vee \ldots \vee A_n \qquad C' \vee \neg A}{(C \vee C')\sigma}$$

provided (i) σ is a most general E-unifier of $A_1 \vee \ldots \vee A_n \vee A$, (ii) no atom in $C\sigma$ is larger or equal to any $A_i\sigma$ with respect to $\prec$, (iii) $\neg A\sigma$ is $\prec$-maximal in $(C' \vee \neg A)\sigma$.

Implicit renaming of variables ensures that the premises are variable disjoint.

A powerful concept is that of redundancy. A ground clause C is *redundant* in a set S (not necessarily ground) with respect to an ordering $\prec$ if there is a (possibly empty) set of ground instances $C_1\sigma_1, \ldots, C_n\sigma_n$ of clauses in S such that

$$\models (C_1\sigma_1 \wedge \ldots \wedge C_n\sigma_n) \rightarrow C$$

and for any i, $C_i\sigma_i \prec C$. A non-ground clause C is *redundant* in S if every ground instance of C is redundant in S.

The left premise of the above resolution rule is referred to as the *positive premise* and the right premise as the *negative premise*. An ordered inference step deriving D from the positive premise C_p and the negative premise C_n is *redundant* in a set S of clauses if S contains clauses $C_{n_1}, \ldots, C_{n_m}$ all strictly smaller than C_n and

$$E \models (C_p \wedge C_{n_1} \wedge \ldots \wedge C_{n_m}) \rightarrow D.$$

Redundancy justifies, for instance, the elimination of tautologies (since $\models \top$) and the replacement of clauses by their normalisations.

Normalisation $\quad S \cup \{C\} \triangleright S \cup \{N(C)\}$

provided $N(C)$ is a normal form of C with the property that (i) $N(C) \preceq C$, and (ii) $N(C)$ and C are logically equivalent under E.

For practical purposes we may stipulate that $N(C)$ can be computed effectively. Condensing is an instance of this rule.

By $\mathcal{R}^{E,\prec}$ we denote any complete resolution procedure employing the above rules. By definition, a set S is *saturated up to redundancy* with respect to $\mathcal{R}^{E,\prec}$, if all inferences with non-redundant premises from S are redundant in S.

Next, we establish (refutational) soundness and completeness for $\mathcal{R}^{E,\prec}$.

Theorem 7.1.1 (Ground completeness of ordered E-resolution) Let $\prec$ be an admissible ordering on ground atoms. Ground ordered E-resolution restricted by $\prec$ with and without refinements compatible with redundancy is a sound and complete resolution procedure.

Proof. By the saturation based construction of Bachmair and Ganzinger (1994, 1997) found also in Appendix B.2. □

For the lifting lemma we need:

Lemma 7.1.2 Let $(\mathcal{A}, \prec)$ be a set of atoms ordered by a (partial) strict ordering. Suppose $\prec$ is stable under the application of substitution. For any two substitutions σ and ρ, if σ is more general than ρ, that is, $\sigma \leq_E \rho[Var(\mathcal{A})]$, and $A\rho$ is $\prec$-maximal in $\mathcal{A}\rho$, then $A\sigma$ is $\prec$-maximal in $\mathcal{A}\sigma$.

Proof. Let θ be a substitution such that $\sigma\theta =_E \rho[Var(\mathcal{A})]$. Then $A\rho$ is maximal in $\mathcal{A}\rho$ iff it is not the case that a B exists in $\mathcal{A}$ such that $A\rho \prec B\rho$, or equivalently, $A\sigma\theta \prec B\sigma\theta$. This implies, by the contrapositive of stability under substitution that, no B exists in $\mathcal{A}$ such that $A\sigma \prec B\sigma$, in other words, $A\sigma$ is maximal in $\mathcal{A}\sigma$. □

An analogous result is true for $(\mathcal{A}/{=_E}, \prec^E)$.

Theorem 7.1.3 (Lifting Lemma) Suppose $\prec$ is an admissible E-compatible ordering on clauses. Let C_1 and C_2 be two variable disjoint clauses, and let C' be a ground $\mathcal{R}^{E,\prec}$-resolvent of the ground clauses $C_1\mu$ and $C_2\nu$. Then there is an $\mathcal{R}^{E,\prec}$-resolvent C of C_1 and C_2 such that $C\theta = C'$ for some substitution θ.

Proof. Assume, on the ground level,

$$C_1\mu = D_1' \vee A_1' \vee \ldots \vee A_n' \quad \text{and} \quad C_2\nu = D_2' \vee \neg A'$$

are such that $\{A_1', \ldots, A_n'\}$ is the largest (with respect to size) subset of atoms occurring in $C_1\mu$ included in $A'/{=_E}$, every atom in D_1' is strictly smaller than any A_i', and $\neg A'$ is maximal in $D_2' \vee \neg A'$. On the non-ground level let

$$C_1 = D_1 \vee A_1 \vee \ldots \vee A_n \quad \text{and} \quad C_2 = D_2 \vee \neg A$$

such that $A_1, \ldots, A_n$ are the atoms in C_1 with $A_i\mu = A_i'$ for any i and $A\nu = A'$. Then, $D_1\mu = D_1'$ and $D_2\nu = D_2'$.

Let $\lambda = \mu \cup \nu$. Then λ is an E-unifier of $A_1 \vee \ldots \vee A_n \vee A$. There is a most general E-unifier σ of $A_1 \vee \ldots \vee A_n \vee A$, and moreover, a unifier θ exists such that $\sigma\theta =_E \lambda[V]$, where V is the set of variables occurring in $A_1 \vee \ldots \vee A_n \vee A$.

We show that the conditions (ii) and (iii) of the resolution rule are true in order to be able to form a resolvent of C_1 and C_2. Let B be any atom in D_1. We show that if $B\sigma/{=_E}$ is comparable with $A\sigma/{=_E}$ then $B\sigma/{=_E} \prec^E A\sigma/{=_E}$. $B\lambda = B\mu = B\sigma\theta$ is a ground atom in D_1' with $B\sigma\theta/{=_E} \prec^E A\sigma\theta/{=_E} = A'/{=_E}$. By Lemma 7.1.2 and since $\prec^E$ is stable under the application of substitution, it follows that $B\sigma/{=_E} \prec^E A\sigma/{=_E}$. Similarly, $\neg A$ is $\prec$-maximal in $D_2 \vee \neg A$.

Consequently, C_1 and C_2 have an $\mathcal{R}^{E,\prec}$-resolvent $C = (D_1 \vee D_2)\sigma$. Thus $C\theta = (D_1 \vee D_2)\sigma\theta = (D_1 \vee D_2)\lambda = D_1\mu \vee D_2\nu = D_1' \vee D_2' = C'$, as required. □

By Theorem 7.1.1 and the Lifting Lemma:

Theorem 7.1.4 (Completeness of ordered E-resolution) Let $\prec$ be an admissible ordering which is compatible with E. Ordered E-resolution by $\mathcal{R}^{E,\prec}$ with and without refinements compatible with redundancy is sound and complete.

As general E-unification is an expensive operation, it may be preferable to use a calculus comprising of at least binary ordered E-resolution and syntactic ordered factoring as in Baumgartner (1992) together with E-normalisation. An advantage of this calculus is that E-unification needs to be applied only to singleton problems, which is less expensive than general E-unification. A disadvantage then, is that for example tautology deletion is no longer compatible. Tautology elimination is especially useful in connection with clauses originating from definitional forms (defined later in Section 7.3).

7.2 Orderings compatible with path theories

The theories we focus on are assumed to be finitely based equational theories E forming convergent rewrite systems oriented by an ordering $\prec$. In general, this implies the sets of rules of the rewrite systems are finite, the E admit *unique normal forms*, and recursive normalisation functions N_E exist. An important consequence for us is that the unique normal forms are the least elements in the E-congruence classes.

In particular, we focus on the equational path theories of the schemas T and 4. The associated rewrite rules are:

$$[x\underline{e}] \Rightarrow x,$$
$$[x(\alpha \circ \beta)] \Rightarrow [x\alpha\beta].$$

They are oriented by the lexicographic path ordering $\prec_{\mathrm{lpo}}$ determined by any total well-founded precedence $\prec$ in which

$$\underline{e} \prec \circ \prec [\cdot,\cdot]. \tag{7.2}$$

The status of $[\cdot,\cdot]$ is right-to-left. Technically this means the permutation π in the definition of the ordering $\prec^{[\cdot,\cdot]}$ is the inversion of the ordering of the arguments of $[\cdot,\cdot]$ (see Appendix B.1). All this means is that, when we compare two paths,

$$[su] \quad \text{and} \quad [s\underline{e}(u \circ (\underline{e} \circ \underline{e}))]$$

say, their ordering is determined by the following alternative encoding

$$f(u, f(s, [])) \quad \text{and} \quad f(u \circ (\underline{e} \circ \underline{e}), f(\underline{e}, f(s, []))),$$

in which $[su]$ corresponds to $f(u,s)$, and according to which $[su] \prec_{\mathrm{lpo}} [s\underline{e}(u \circ (\underline{e} \circ \underline{e}))]$.

Keep in mind, the normal forms defined by N_T, N_4 and N_{T4} are basic paths, that is, terms of the kind $[u_1 \ldots u_n]$, where each u_i is either a functional variable or a functional constant.

Now, we define the extensions of $\prec_{\mathrm{lpo}}$ to atoms, literals and clauses. As is common, we denote all these by the same symbol, namely $\prec$. The ordering on ground atoms is the lexicographic path ordering determined by the extension of the precedence (7.2) with predicate symbols having lower precedence than function symbols. This implies the ordering on ground atoms is determined first by the ordering on the arguments and then on the ordering of the predicate symbols: $Ps \prec Qt$ iff (i) $s \prec t$ or (ii) $s = t$ and $P \prec Q$. We assume the ordering on literals is the multi-set ordering on a multi-set encoding of any atom A by $\{A\}$ and any negated atom $\neg A$ by $\{A, A\}$ (there are other ways of defining an ordering on ground literals). The ordering on ground clauses is the induced multi-set ordering. The corresponding orderings on non-ground expressions are defined in analogy to the following for atoms: for any non-ground atoms A and B

$$A \prec B \quad \text{iff} \quad \text{for any ground substitution } \sigma,\ A\sigma \prec B\sigma.$$

The described orderings are admissible.

It depends on the theory E whether or not $\prec$ is compatible with E. For the empty theory the question is irrelevant. By the general completeness result of the previous section (Theorem 7.1.4) the following is immediate.

Corollary 7.2.1 Let $\prec$ be the lexicographic path ordering on atoms induced by any precedence on the basic vocabulary. Then, $\mathcal{R}^{\prec}$ and $\mathcal{R}^{\prec}_{\text{COND}}$ are complete resolution procedures for the basic path logic.

Finding a compatible ordering for non-empty E is less straightforward. Assuming that $\underline{e} \prec \underline{\alpha} \prec \underline{\beta} \prec \ldots \prec \circ \prec [\cdot,\cdot]$ for path logics with associativity and identity we have that

$$P[\underline{\alpha}] \prec P[\underline{\alpha}\underline{\beta}],$$

regardless of whether $\underline{\alpha} \prec \underline{\beta}$ or $\underline{\beta} \prec \underline{\alpha}$. When $\underline{\beta} \prec \underline{\alpha}$ then

$$P[\underline{\alpha}\underline{\beta}] \prec P[\underline{e}(\underline{\alpha} \circ (\underline{e} \circ \underline{e}))],$$

implying that $P[\underline{\alpha}]/{=_E}$ and $P[\underline{\alpha}\underline{\beta}]/{=_E}$ are not comparable by $\prec^E$, and hence $\prec$ is not compatible with associativity and identity. The ordering $\prec$ is also not compatible with associativity alone as demonstrated by this example.

$$P[\underline{\alpha}\underline{\beta}] \prec P[\underline{\alpha}\underline{\beta}\underline{\gamma}], \quad \text{but} \quad P[\underline{\alpha}\underline{\beta}\underline{\gamma}] \prec P[\underline{\alpha} \circ \underline{\beta}],$$

when the constants are smaller than $\circ$ and also when $\circ \prec \underline{\gamma} \prec \underline{\alpha}$ (since $\underline{\gamma} \prec \underline{\alpha} \circ \underline{\beta}$ and thus $f(\underline{\gamma}, f(\underline{\beta}, f(\underline{\alpha}, []))) \prec f(\underline{\alpha} \circ \underline{\beta}, [])$, in both cases).

We will now define a compatible ordering $\prec^{\times}$, without requiring special precedences on constants. Suppose $\prec$ is based on the precedence (7.2) with predicate symbols being smaller than function symbols. Define $(\mathcal{A}, \prec^{\times})$ by the lexicographic combination of the orderings $(N_E(\mathcal{A}), \prec)$ and $(\mathcal{A}, \prec)$. More specifically, for any $A, B \in \mathcal{A}$, let

$$A \prec^{\times} B \quad \text{iff} \quad (N_E(A), A) \prec (N_E(B), B),$$

that is, (i) $N_E(A) \prec N_E(B)$ or (ii) $N_E(A) = N_E(B)$ and $A \prec B$. The idea is that the induced ordering $\prec^{\times,E}$ of the congruence classes mirrors the total ordering by $\prec$ of the unique normal forms.

Lemma 7.2.2 If the extension of $(\mathcal{A}, \prec)$ to ground literals is admissible then so is the extension of $(\mathcal{A}, \prec^{\times})$ to ground literals.

Proof. It is not difficult to verify that the operation $\cdot^{\times}$ preserves being a total and well-founded strict ordering, and the properties that $A \prec \neg A$ and if $A \prec B$ then $\neg A \prec B$ remain invariant under $\cdot^{\times}$. □

Lemma 7.2.3 Let E be an equational path theory which forms a convergent rewrite system oriented by $\prec$. If $(\mathcal{A}, \prec)$ is a total well-founded and strict ordering on ground atoms then so is $(\mathcal{A}/{=_E}, \prec^{\times,E})$.

Proof. We just show that every pair of distinct X and Y in $\mathcal{A}/{=_E}$ are comparable by $\prec^{\times,E}$, and leave the remainder for the reader to fill in. Let $A \in X$ and $B \in Y$ be arbitrary. $N_E(A)$ and $N_E(B)$ are the least elements of $(X, \prec)$ and $(Y, \prec)$, respectively. That is, $N_E(A) \preceq A$ and $N_E(B) \preceq B$. Without loss of generality, assume $N_E(A) \prec N_E(B)$. Hence $A \prec^{\times} B$. □

This proves the ordering $\prec^{\times}$ on sets of ground expressions is compatible with E.

As before, on the non-ground level the ordering $\prec^{\times}$ is defined by, $A \prec^{\times} B$ iff for any ground substitution σ, $A\sigma \prec^{\times} B\sigma$, that is, $(N_E(A\sigma), A\sigma) \prec (N_E(B\sigma), B\sigma)$.

The ordering $\prec^{\times}$ is a suitable ordering (Lemmas 7.2.2 and 7.2.3) giving us complete ordered E-resolution procedure for path logics satisfying identity and/or associativity (by Theorem 7.1.4). Like condensing the relevant normalisation functions N_E are instances of the general normalisation rule of the previous section. Also covered is semantic condensation. The *semantic condensation* of a clause C is the smallest subset C' of C such that $C' = N_E(C\sigma)$[2] for some substitution σ. Therefore:

Theorem 7.2.4 Ordered E-resolution restricted by $\prec^{\times}$ with and without redundancy eliminating refinements (in particular, $\mathcal{R}^{E,\prec^{\times}}$, $\mathcal{R}^{E,\prec^{\times}}_{N_E}$, $\mathcal{R}^{E,\prec^{\times}}_{\mathrm{COND}}$ and $\mathcal{R}^{E,\prec^{\times}}_{\mathrm{COND}\,\circ\, N_E}$) are complete resolution procedures for path logics with $E \subseteq \{\text{associativity}, \text{identity}\}$.

Finally we observe, for the theories under consideration $\prec^{\times}$ is determined by the proper subterm ordering. More precisely:

Lemma 7.2.5 Under the empty theory or associativity the ordering $\prec^{\times}$ among (ground or non-ground) atoms in normal form is determined by: $Ps \prec^{\times} Qt$ provided (i) s is a proper subterm of t, or (ii) $s = t$ and $P \prec Q$.

Proof. For $E = \emptyset$, N_E is the identity mapping and $Ps \prec^{\times} Qt$ iff $Ps \prec Qt$. It is not difficult to prove: $s \prec t$ iff s is a proper subterm of t, when s and t are basic paths.

For $E = \{\text{associativity}\}$, instantiation with complex functional terms of the form $u \circ v$ preserves the orderings. That is, if s is a proper subterm of t then $s\sigma \prec^{\times} t\sigma$, for any substitution σ. □

This is not the case under collapsing axioms. For example, under identity

$P[\alpha] \prec^{\times} Q[\alpha\beta]$	only when $P \prec Q$, but
$P[\alpha]$ and $Q[\alpha\beta]$ are not comparable	when $Q \prec P$,

since $N_E(Q[\underline{\alpha}\underline{e}]) = Q[\underline{\alpha}] \prec P[\underline{\alpha}]$ and $P[\underline{\alpha}] \prec Q[\underline{\alpha}\underline{\beta}]$. We need an additional restriction in (i):

Lemma 7.2.6 Under identity the ordering $\prec^{\times}$ among (ground or non-ground) atoms in normal form is determined by: $Ps \prec^{\times} Qt$ iff (i) s is a proper subterm of t and $P \prec Q$, or (ii) $s = t$ and $P \prec Q$.

[2] Here, $N_E(C\sigma)$ must be viewed as a set.

7.3 Modal definitional forms

A special form of clausal form often used in automated theorem proving is the *(clausal) definitional forms* obtained by *renaming* (Plaisted and Greenbaum 1986, Boy de la Tour 1992, Eder 1992). Tseitin (1970) introduced definitional forms for propositional formulae for efficiency reasons. The conventional transformation of a given (propositional or first-order) formula φ to clausal form does conversion to conjunctive or negation normal form followed by Skolemisation. This causes an exponential size increase, because it uses distributivity laws. In contrast, the conversion to definitional form by renaming has linear time complexity and causes merely a linear increase in size. In point of fact, for any endeavour of facilitating modal deduction by resolution type inference, the efficiency argument is decisive. Except for *S5*, most modal logics we consider in this thesis are PSPACE-complete and a natural prerequisite is that the complexity of the reduction of a modal formula to clausal form should not exceed that of testing satisfiability for the given logic. A second reason why definitional forms are often preferred is that the structure of the original formula is retained in the clausal form provided definitions are introduced for every non-literal subformula. For similar reasons modal definitional forms are being used in modal resolution calculi as described in Mints (1989, 1990). A pleasant side effect for us is that, definitional forms give us more control over the Skolem terms (constants in our situation).

We need some notation for identifying occurrences of subformulae in a given formula φ. Let O be a set of elements identifying occurrences (or positions) in a formula.[3] When writing $\varphi[\psi_1]_O$ we assume O is a non-empty set of occurrences of ψ_1 in φ. By $\varphi[\psi_1 \mapsto \psi_2]_O$ we denote the formula obtained from $\varphi[\psi_1]_O$ by replacing every occurrence in O of ψ_1 by ψ_2.

The fundamental idea of renaming can be explained best on propositional formulae. The method introduces new propositional symbols for subformulae of the given φ and exploits the replacement of equivalents property (and possibly the monotonicity laws of $\rightarrow$ and $\leftarrow$). For propositional logic the replacement of equivalents property is:

$$(\psi_1 \leftrightarrow \psi_2) \rightarrow (\varphi[\psi_1]_O \leftrightarrow \varphi[\psi_1 \mapsto \psi_2]_O). \tag{7.3}$$

It says, the formula $\varphi[\psi_1 \mapsto \psi_2]_O$ is obtained from $\varphi[\psi_1]_O$ by replacing every occurrence in O of ψ_1 by its logical equivalent ψ_2. In particular, if we let a be a new propositional variable and ψ a subformula of φ then

$$(\psi \leftrightarrow a) \rightarrow (\varphi[\psi]_O \leftrightarrow \varphi[\psi \mapsto a]_O).$$

A definitional form of a propositional formula φ is obtained by the repeated application of the following rule:

$$\varphi \quad \text{becomes} \quad (\psi \leftrightarrow a) \rightarrow \varphi[\psi \mapsto a]_O,$$

for O a subset of occurrences of ψ in φ and a a new propositional variable not occurring in φ. $\psi \leftrightarrow a$ is called the *definition* or *defining part* of ψ, and we say *a defines ψ*. Conversion to definitional form does not preserve logical equivalence, but we have: Any definitional form of a formula is valid iff the formula is valid. Equivalently, φ is (un)satisfiable iff so is

[3] Usually occurrences are sequences of natural numbers that identify nodes in the syntactic tree representation of a formula.

$(\psi \leftrightarrow a) \wedge \varphi[\psi \mapsto a]_O$. We will prove a corresponding result for modal definitional forms below.

Consider an example. For the formula

$$\rho_4 = \underset{a_1}{(p \to q)} \underset{a_3}{\to} \underset{a_2}{(\neg p \wedge q)}$$

we introduce the new propositional names a_1, a_2, a_3 as indicated, a_1 for $p \to q$, a_2 for $\neg p \wedge q$ and a_3 for $a_1 \to a_2$ (other choices are possible), and get the definitional form

$$\rho_4^d = [(a_1 \leftrightarrow (p \to q)) \wedge (a_2 \leftrightarrow (\neg p \wedge q)) \wedge (a_3 \leftrightarrow (a_1 \to a_2)] \to a_3.$$

$\neg\rho_4$ is unsatisfiable, which we prove by refuting $\neg\rho_4^d$. Now, note that each definition $a_i \leftrightarrow \psi$ is a set of clauses with at most three literals. Evidently, the conversion of $\neg\rho^d$ to the clausal form $c(\neg\rho^d)$ has linear complexity both with regards time and space.

Definitional forms are not unique, and there are different forms of renaming that are utilised in different situations. We have used a common strategy of iteratively introducing new names for all smallest subformulae that are not literals. Advanced forms of renaming are discussed in Plaisted and Greenbaum (1986) and Rock (1995) and are implemented in the theorem prover SPASS (Weidenbach, Gaede and Rock 1996, Weidenbach 1997). A sensible simplification is to use the replacement of equivalents property sparingly, and instead use the monotonicity property of $\to$ or converse monotonicity property of $\leftarrow$, for positive and negative context application, respectively. Namely:

$$(\psi_1 \to \psi_2) \to (\varphi[\psi_1]_O \to \varphi[\psi_1 \mapsto \psi_2]_O)$$

provided the context of the occurrences in O of ψ_1 are all positive, and

$$(\psi_1 \leftarrow \psi_2) \to (\varphi[\psi_1]_O \to \varphi[\psi_1 \mapsto \psi_2]_O)$$

provided the context of the occurrences in O of ψ_1 are all negative. An occurrence of a subformula has *positive polarity* if it is one inside the scope of an even number of (explicit or implicit) negations, and an occurrence has *negative polarity* if it is one inside the scope of an odd number of negations. For example, above in ρ_4 the subformula $\neg p \wedge q$ has positive polarity, $p \to q$ has negative polarity, and both φ and ψ in $\varphi \leftrightarrow \psi$ have neither positive nor negative polarity regardless of the polarity of the double implication. The appropriate optimised reduction rules are then

$$\varphi \quad \text{becomes} \quad D(a, \psi) \to \varphi[\psi \mapsto a]_O$$

and $D(a, \psi)$ denotes the definition of ψ given by

$\psi \to a$,	provided every occurrence in O of ψ has positive polarity
$\psi \leftarrow a$,	provided every occurrence in O of ψ has negative polarity, and
$\psi \leftrightarrow a$,	otherwise.

For the sample formula ρ_4 an optimised definitional form is

$$[(a_1 \to (p \to q)) \wedge (a_2 \leftarrow (\neg p \wedge q)) \wedge (a_3 \leftarrow (a_1 \to a_2)] \to a_3$$

because it produces a smaller clause set.

In first-order logic the replacement of equivalents and monotonicity properties are context sensitive. A definitional form of a first-order formula φ is obtained by the repeatedly applying

$$\varphi \quad \text{becomes} \quad D(A,\psi) \to \varphi[\psi \mapsto A]_O,$$

where $D(A,\psi)$ denotes the definition of ψ and depending on the polarity of the occurrence of ψ, it is the universal closure of $\psi \to A$, $\psi \leftarrow A$ or $\psi \leftrightarrow A$. The symbol A denotes an atom $P(x_1,\ldots,x_k)$ for P a new predicate symbol and $x_1,\ldots,x_k$ are the free variables of ψ.

Like in first-order logic, in modal logic the context is important. For example, the replacement of equivalents for normal modal logics is:

$$\Box^{(n)}(\psi_1 \leftrightarrow \psi_2) \to (\varphi[\psi_1]_O \leftrightarrow \varphi[\psi_1 \mapsto \psi_2]_O),$$

where n is the maximum modal depth of any occurrence in O of ψ_1. $\Box^{(n)}(\psi_1 \leftrightarrow \psi_2)$ denotes the 'modal universal closure' of the formula $\psi_1 \leftrightarrow \psi_2$ and is defined by

$$\Box^{(n)}\psi = \psi \wedge \Box\psi \wedge \Box^2\psi \wedge \ldots \wedge \Box^n\psi,$$

for n a non-negative integer. $\Box^n$ denotes a string of n box operators, formally, $\Box^0\varphi = \varphi$ and $\Box^n\varphi = \Box\Box^{n-1}\varphi$ for $n > 0$. Intuitively, if $\Box^{(n)}\varphi$ is true in a world x then φ is true in x and all worlds reached from x by n steps.

We have some freedom in defining modal definitional forms and the definition we give now is very general. For a given modal formula φ a *(modal) definitional form* of φ, denoted by φ^d, is obtained by repeatedly introducing for any subformula ψ of φ a new propositional variable a and transforming φ to the formula

$$D(a,\psi) \to \varphi[\psi \mapsto a]_O$$

with

$$D(a,\psi) = \begin{cases} \Box^{(n)}(\psi \to a) & \text{provided every occurrence in } O \text{ of } \psi \text{ has positive polarity} \\ \Box^{(n)}(\psi \leftarrow a) & \text{provided every occurrence in } O \text{ of } \psi \text{ has negative polarity, and} \\ \Box^{(n)}(\psi \leftrightarrow a) & \text{otherwise,} \end{cases}$$

and n is the modal depth of the occurrence ψ in φ that is replaced by a, or if we replace more than one occurrence by a, n is the maximal modal depth. Peculiar to modal logic, we may be more economical and simplify $\Box^{(n)}$ in the defining part to $\Box^{(n_i)_i}$, if $(n_i)_i$ is the sequence $n_1,\ldots,n_m$ of modal depths of the occurrences of ψ that are replaced by a. Then, $\Box^{(n_i)}\varphi = \Box^{n_1}\varphi \wedge \ldots \wedge \Box^{n_m}\varphi$. The Preview gives a non-optimal definitional form of

$$\rho_3 = \underset{a_1}{\Box p} \to \underset{a_3 a_2 a_1}{\Box\,\Box\,\Box p}. \tag{7.4}$$

Another definitional form is

$$\begin{aligned} \rho_3^d = [&(a_1 \leftrightarrow \Box p) \wedge \Box^2(a_1 \leftrightarrow \Box p) \wedge \\ &\Box(a_2 \leftarrow \Box a_1) \wedge (a_3 \leftarrow \Box a_2)] \to (a_1 \to a_3). \end{aligned}$$

A formula is not logically equivalent to its definitional form, but:

Theorem 7.3.1 Let φ be any modal formula in $K_{(m)}\Sigma$ and let φ^d be a definitional form of φ. Then

$$\varphi \text{ is a theorem in } K_{(m)}\Sigma \quad \text{iff} \quad \text{so is } \varphi^d.$$

Proof. It suffices to prove the result for one step of the transformation procedure. Let ψ be any subformula of φ and let a be a new propositional variable not occurring in φ. To avoid cluttering we write $\varphi[\psi]$ instead of $\varphi[\psi]_O$ and $\varphi[a]$ instead of $\varphi[\psi \mapsto a]_O$. We prove:

$$(') \qquad \varphi[\psi] \text{ is a theorem} \quad \text{iff} \quad \Box^{(n)}(a \leftrightarrow \psi) \to \varphi[a] \text{ is a theorem.}$$

The following, in essence, replacement of equivalents, is a theorem in $K_{(m)}\Sigma$.

$$('') \qquad \Box^{(n)}(\psi \leftrightarrow a) \to (\varphi[\psi] \leftrightarrow \varphi[a])$$

(The proof is by induction on the structure of $\varphi[\psi]$.)[4] For the ($\Rightarrow$) direction of ($'$) suppose φ holds (is a theorem) in $K_{(m)}\Sigma$. Assume $\Box^{(n)}(\psi \leftrightarrow a)$ holds, too. Applying modus ponens twice, first with ($''$) and then with $\varphi[\psi] \to \varphi[a]$, we derive that $\varphi[a]$ holds. Discarding the assumption that $\Box^{(n)}(\psi \leftrightarrow a)$ holds, we conclude that $\Box^{(n)}(\psi \leftrightarrow a) \to \varphi[a]$ is a theorem of $K\Sigma$.

For the ($\Leftarrow$) direction assume $\Box^{(n)}(\psi \leftrightarrow a) \to \varphi[a]$ holds. Then, using ($''$),

$$\Box^{(n)}(\psi \leftrightarrow a) \to (\varphi[a] \wedge (\varphi[a] \leftrightarrow \varphi[\psi]))$$

holds, which implies $\Box^{(n)}(\psi \leftrightarrow a) \to (\varphi[a] \to \varphi[\psi]))$ and this is equivalent to

$$(\Box^{(n)}(\psi \leftrightarrow a) \to \varphi[a]) \to \varphi[\psi].$$

An application of modus ponens yields $\varphi[\psi]$ as required.

The proof(s) of: $\varphi[\psi]$ is a theorem iff $\Box^{(n)}(\psi \to a) \to \varphi[a]$ is a theorem (respectively, $\Box^{(n)}(\psi \leftarrow a) \to \varphi[a]$ is a theorem), for when the occurrences in O of ψ all have positive polarity (respectively, negative polarity) are similar. $\Box$

In some modal logics definitional forms have simpler forms, in particular, a simpler context. The next result lists some simplifications for the defining part $D(a, \psi)$ of different modal logics.

Theorem 7.3.2 Let φ be any modal formula with subformula ψ occurring in the scope of maximally n modal operators. In extensions of K that include the following schemas the defining part $D(a, \psi) = \Box^{(n)}(\psi \mathbin{@} a)$, for @ either $\leftrightarrow$, $\to$ or $\leftarrow$, is equivalent to the appropriate formula in the right column of the following table:

$T = \Box p \to p$	$\Box^n(\psi \mathbin{@} a)$
$4 = \Box p \to \Box\Box p$	$(\psi \mathbin{@} a) \wedge \Box(\psi \mathbin{@} a)$
$T, 4$	$\Box(\psi \mathbin{@} a)$

Proof. The proof is routine. For example, when a logic includes the schema T then $\Box^n\psi$ implies $\Box^m\psi$ for all $0 \leq m < n$. $\Box$

[4]The proof for the analogue property of first-order logic can be found in Mendelson (1987, Proposition 2.8).

In general, it makes a difference if we introduce the definitional form in the modal context or if we postpone it to after the conversion to first-order logic. For example, ρ_3 translates by Π_f in KD to

$$\begin{array}{ll} \forall\alpha P[\alpha] \rightarrow & \forall\alpha\forall\beta\forall\gamma P[\alpha\beta\gamma] \\ A_1 & A_3 A_2 A_4 \end{array}$$

Above in (7.4) we replaced both occurrences of $\Box p$ by the new symbol a_1, but evidently the corresponding replacement is not possible here. On the first-order level four names would be introduced as indicated.

It is not difficult to see, transformation of modal formulae to definitional form (that avoids the application of any distributive laws) followed by the optimised functional translation to clausal form of basic path logic can be achieved by a linear time algorithm.

7.4 Prefix ordered clauses and decidability of ordered resolution

This section studies two renaming methods. The resulting clausal forms have the property that the terms in any clause are ordered by the subterm ordering. We call clauses with this property *prefix ordered*. We will see, prefix ordering is preserved by ordered resolution for the empty theory and associativity.

Initially, we assume that names are introduced systematically for *modally quantified* propositional subformulae. More precisely, we assume any modal formula φ has been converted to a form

$$\varphi^d = D_\varphi \rightarrow \varphi',$$

where

(i) D_φ is the conjunction of all defining parts $D(a, \psi)$,

(ii) ψ has the form $\mathbf{Q}_1 \ldots \mathbf{Q}_k \psi'$ for ψ' a propositional formula and $\mathbf{Q}_1 \ldots \mathbf{Q}_k$ a sequence of modal operators, and

(iii) φ' is a propositional formula.

Consider a sample formula $\rho_5 = \Diamond(\neg\Diamond\Box(\neg p \wedge \neg q) \wedge r)$. A definitional form that satisfies the assumptions is

$$\rho_5^d = D_{\rho_5} \rightarrow \rho_6' = [\Box(a_1 \leftrightarrow \Diamond\Box(\neg p \wedge \neg q)) \wedge (a_2 \leftrightarrow \Diamond(\neg a_1 \wedge r))] \rightarrow a_2. \tag{7.5}$$

Using the translation for serial modalities, the set $S = c^\Upsilon(\neg\Pi_f(\varphi^d))$ consists of the following clauses:

1. $\neg A_1[\alpha] \vee \neg P[\alpha\gamma\underline{\beta}]$	5. $\neg A_2[] \vee A_1[\underline{\delta}]$
2. $\neg A_1[\alpha] \vee \neg Q[\alpha\gamma\underline{\beta}]$	6. $\neg A_2[] \vee \neg R[\underline{\delta}]$
3. $A_1[\alpha] \vee P[\alpha\gamma\beta] \vee Q[\alpha\gamma\beta]$	7. $A_2[] \vee \neg A_1[\delta] \vee R[\delta]$
4. $\neg A_2[]$.	

The set of terms of any clause is ordered by the proper subterm ordering, called the prefix ordering.

Formally, we say a clause is *prefix ordered* if it is a clause of basic path logic and has the form:

$$E_0 s \vee E_1[su_1] \vee E_2[su_1u_2] \vee \ldots \vee E_n[su_1 \ldots u_n]. \tag{7.6}$$

Each $E_i[su_1 \ldots u_i]$ denotes a (finite and possibly empty) disjunction

$$L_1[su_1 \ldots u_i] \vee \ldots \vee L_k[su_1 \ldots u_i]$$

of literals with the same argument $[su_1 \ldots u_i]$. s may be the empty path. (The notation $L_i s$ makes the argument s of the literal L_i explicit.) Evidently, a clause is prefix ordered iff the set of its terms is ordered totally by the proper subterm ordering.

We will now present a general characterisation of clausal forms $c^{\Upsilon}(\neg\Pi_f(\varphi^d))$ that originate from definitional forms and satisfy (i), (ii) and (iii) above. Recall that, $\neg\Pi_f(\varphi) = \exists x\ \neg\pi_f(\varphi, x)$ and the empty string is the Skolem constant associated with $\exists x$. Then $S = c^{\Upsilon}(\neg\Pi_f(\varphi^d)) = c^{\Upsilon}(\neg\Pi_f(D_\varphi \to \varphi'))$ is given by

$$c^{\Upsilon}(\pi(D_\varphi, [])) \ \cup \ c^{\Upsilon}(\neg\pi(\varphi', [])). \tag{7.7}$$

Lemma 7.4.1 The set $c^{\Upsilon}(\neg\pi(\varphi', []))$ is a set of clauses of the form

$$L_1[] \vee \ldots \vee L_k[],$$

and each clause in $c^{\Upsilon}(\neg\pi(\varphi', []))$ is ground and prefix ordered.

Proof. Since φ' is a propositional formula the paths in its translation are all empty and $c^{\Upsilon}(\neg\pi(\varphi', []))$ is a set of clauses of unary literals of the form $\pm P[]$. (In the example, φ' is a_2 and $c^{\Upsilon}(\neg\pi(\varphi', [])) = \{\neg A_2[]\}$.) $\square$

Lemma 7.4.2 Let C be any clause in $c^{\Upsilon}(\pi(D_\varphi, []))$. For serial modal logics, C has the form

$$\pm A[\alpha_1 \ldots \alpha_n] \vee L_1[\alpha_1 \ldots \alpha_n u_1 \ldots u_k] \vee \ldots \vee L_m[\alpha_1 \ldots \alpha_n u_1 \ldots u_k],$$

and for non-serial modal logics, C has the form

$$\pm A[\alpha_1 \ldots \alpha_n] \vee E_0[\alpha_1 \ldots \alpha_n] \vee E_1[\alpha_1 \ldots \alpha_n u_1] \vee \ldots \vee E_k[\alpha_1 \ldots \alpha_n u_1 \ldots u_k],$$

for $n \geq 0$ and $m, k > 0$. In both cases, C is prefix ordered.

In the example, clauses 1. to 3. originate with the definition $\Box(a_1 \leftrightarrow \Diamond(\neg p \wedge \neg q))$ and clauses 5. to 7. originate with $a_2 \leftrightarrow \Diamond(\neg a_1 \wedge r)$.

Proof. We will verify only the serial case. For the non-serial case we must also consider the additional dead-end expressions.

D_φ is a conjunction of definitions of the form

$$\Box^n(a \leftarrow \mathbf{Q}_1 \ldots \mathbf{Q}_k \psi'), \quad \Box^n(a \to \mathbf{Q}_1 \ldots \mathbf{Q}_k \psi') \quad \text{or} \quad \Box^n(a \leftrightarrow \mathbf{Q}_1 \ldots \mathbf{Q}_k \psi') \tag{$'$}$$

for ψ' a propositional formula and $\mathbf{Q}_1 \ldots \mathbf{Q}_k$ a sequence of modal operators. $\Box^n(a \leftrightarrow \psi)$ is equivalent to $\Box^n(\neg a \vee \psi) \wedge \Box^n(a \vee \neg\psi)$ (because $\Box$ distributes over conjunction). By the functional translation for serial modal logics, the formulae in $(')$ transform to one of the following or their conjunction

$$('') \qquad \begin{array}{l} \forall\alpha_1 \ldots \forall\alpha_n\ (\neg A[\alpha_1 \ldots \alpha_n] \vee Q_1 u_1 \ldots Q_k u_k\ \psi'') \\ \forall\alpha_1 \ldots \forall\alpha_n\ (A[\alpha_1 \ldots \alpha_n] \vee Q_1^{\neg} u_1 \ldots Q_k^{\neg} u_k\ \psi''^{\neg}). \end{array}$$

The modal prefix $\Box^n$ becomes a prefix of universal quantifiers $\forall\alpha_1 \ldots \forall\alpha_n$, the propositional variable a becomes a unary predicate variable A, and ψ and $\neg\psi$ become complementary formulae $Q_1 u_1 \ldots Q_k u_k\ \psi''$ and $Q_1^{\neg} u_1 \ldots Q_k^{\neg} u_k\ \psi''^{\neg}$. The terms in ψ'' and $\psi''^{\neg}$ are identical, they are $[\alpha_1 \ldots \alpha_n u_1 \ldots u_k]$. Each Q_i denotes either a universal quantifier or an existential quantifier depending on whether $\mathbf{Q}_i$ is a box or a diamond operator and $Q_i^{\neg}$ denotes the dual quantifier of Q_i. It follows that the clauses in the optimised clause form of $('')$ have the required form. $\Box$

Theorem 7.4.3 Let φ be any multi-modal formula. Any clause in $S = c^{\Upsilon}(\neg\Pi_f(\varphi^d))$ is prefix ordered.

Proof. By Theorem 5.1.2 (S is well-formed in path logics) and Lemmas 7.4.1 and 7.4.2. $\Box$

The second form of renaming we consider is an instance of the first. Renaming is done for any non-literal subformula, that is, the conditions (ii) and (iii), above, are replaced by

(ii) ψ has the form ψ' @ ψ'' for @ a Boolean connective, $\mathbf{Q}\psi'$ for $\mathbf{Q}$ either a $\Box$ or a $\Diamond$ operator, and ψ' and ψ'' are propositional atoms or negations of atoms, and

(iii) φ' is a propositional atom or literal.

The following describes the clausal form.

Corollary 7.4.4 Clauses in $S = c^{\Upsilon}(\neg\Pi_f(\varphi^d))$ have the form

$$\begin{array}{l} L_1[\alpha_1 \ldots \alpha_n] \vee L_2[\alpha_1 \ldots \alpha_n] \\ L_1[\alpha_1 \ldots \alpha_n] \vee L_2[\alpha_1 \ldots \alpha_n] \vee L_3[\alpha_1 \ldots \alpha_n] \\ L_1[\alpha_1 \ldots \alpha_n] \vee L_2[\alpha_1 \ldots \alpha_n u] \\ L_1[\alpha_1 \ldots \alpha_n] \vee L_2[\alpha_1 \ldots \alpha_n] \vee L_3[\alpha_1 \ldots \alpha_n u] \\ L[] \end{array}$$

with $0 \leq n \leq m$ (for some m), u a variable or a constant, and all clauses are prefix ordered.

Proof. The first two clauses stem from definitions of the propositional expressions ψ' @ ψ'', The next four from definitions of modal expressions $\mathbf{Q}\psi'$, and the final clause originates from the literal φ'. $\Box$

The three literal clauses for the modally quantified formula $\mathbf{Q}\psi'$ emanate in non-serial contexts (where the dead-end literals keep track of defined-ness) and can be avoided by either translation according to Section 2.6 into a serial context or by renaming on the first-order level (after translation by $\Upsilon\Pi_f$).

Ordinary unrestricted resolution does not preserve the prefix ordering. A sample derivation on the sample set S for ρ_5^d in (7.5) is:

$$8.\ \neg A_2[] \vee \neg P[\underline{\delta}\gamma\underline{\beta}] \qquad [1, 5, \text{empty } E]$$
$$9.\ \neg A_1[\delta] \vee R[\delta] \vee \neg P[\underline{\delta}\gamma\underline{\beta}]. \qquad [7, 8, \text{empty } E]$$

Clause 9. is not prefix ordered. (Of course, clause 9. can be split into prefix ordered subclauses, but for example $\neg A_1[\alpha\delta] \vee R[\alpha\delta] \vee \neg P[\alpha\underline{\delta}\gamma\underline{\beta}]$ cannot.)

For theories not involving collapsing axioms, like the empty theory and the theory containing just associativity, prefix ordering is preserved by ordered E-resolution with normalisation using $\mathcal{R}_{N_E}^{E,\prec^\times}$ or $\mathcal{R}_{\mathrm{COND}\circ N_E}^{E,\prec^\times}$. This is due to the fact that there is a correlation between $\prec^\times$ and the proper subterm ordering for associativity (Lemma 7.2.5), but not for the collapsing axioms. For example, when A_2 the largest predicate symbol occurring in the two clauses

$$1.\ \neg A_2[\alpha_1] \vee \neg P[\alpha_1\alpha_2]$$
$$2.\ \ A_2[\alpha_1] \vee \neg A_1[\alpha_1\alpha_2] \vee R[\alpha_1\alpha_2],$$

then an ordered resolvent with respect to $\prec^\times$ and modulo identity is

$$3.\ \ P[\alpha_1\alpha_2] \vee \neg A_1[\alpha_1\beta] \vee R[\alpha_1\beta], \qquad [1, 2, \text{identity}]$$

which is not prefix ordered. Consequently, the following results apply only to the empty theory and associativity. Let $\lessdot$ denote the proper subterm ordering.

Lemma 7.4.5 Let σ be a most general E-unifier of a subset T' of a set T of prefix ordered terms.

(i) If σ is a syntactic most general unifier, then $(T\sigma, \lessdot)$ is prefix ordered.

(ii) If σ is a most general unifier modulo associativity, then $(N_E(T\sigma), \lessdot)$ is prefix ordered.

Proof. Because the proper subterm ordering $\lessdot$ remains invariant under the application of any ground substitution σ and $N_4(s)\sigma \lessdot N_4(s)\sigma$ implies $N_4(s\sigma) \lessdot N_4(s\sigma)$. □

Therefore, for the empty theory or associativity, condensing and factoring (including both syntactic and semantic factoring modulo associativity followed by normalisation) preserves the prefix ordering. For preservation of the ordering in resolvents we need:

Lemma 7.4.6 Let $(T, \lessdot)$ and $(T', \lessdot)$ be two finite variable disjoint and prefix ordered chains of terms. Let E be either empty or let E contain just associativity. Suppose σ is an idempotent most general E-unifier of t and t' and $N_E(t\sigma) = N_E(t'\sigma)$ are maximal in $N_E(T\sigma)$ and $N_E(T'\sigma)$, respectively. Then $(N_E((T \cup T')\sigma), \lessdot)$ is a prefix ordered chain.

Proof. We must show that for any $s \in T$ and any $s' \in T'$, $N_E(s\sigma)$ and $N_E(s'\sigma)$ coincide or are related by $\lessdot$. Both are prefixes of $N_E(t\sigma) = N_E(t'\sigma)$, the maximal element in $N_E((T \cup T')\sigma)$. Then $[N_E(s\sigma)r] = [N_E(s'\sigma)r']$ for some strings r and r'. If $N_E(s\sigma)$ is shorter than $N_E(s'\sigma)$, then $N_E(s\sigma) \lessdot N_E(s'\sigma)$, otherwise we have that $N_E(s'\sigma) \lessdot N_E(s\sigma)$. □

The two lemmas together with Lemma 7.2.5 imply:

Theorem 7.4.7 Any resolvent of two prefix ordered basic path clauses by $\mathcal{R}^{\prec}$ and $\mathcal{R}^{\prec}_{\text{COND}}$ modulo the empty theory and any resolvent by $\mathcal{R}^{4,\prec^\times}_{N_4}$ or $\mathcal{R}^{4,\prec^\times}_{\text{COND}\,\circ\,N_4}$ modulo associativity, is prefix ordered.

Hence:

Theorem 7.4.8 Let φ be any modal formula and $S = c^{\Upsilon}(\neg\Pi_f(\varphi^d))$ with φ^d defined as in this section. For any n, any clause in $(\mathcal{R}^{\prec})^n(S)$, $(\mathcal{R}^{\prec}_{\text{COND}})^n(S)$, $(\mathcal{R}^{4,\prec^\times}_{N_4})^n(S)$ or $(\mathcal{R}^{4,\prec^\times}_{\text{COND}\,\circ\,N_4})^n(S)$ is prefix ordered.

For the empty theory, a term depth bound exists and decidability follows with little effort for ordered resolution, as we will see now (without relying on Theorem 6.5.1). Evidently, the number of variables in any prefix ordered clause

$$E_0 s \vee E_1[su_1] \vee E_2[su_1u_2] \vee \ldots \vee E_n[su_1 \ldots u_n]$$

is given by the number of variables in largest term occurring in the clause, namely $[su_1 \ldots u_n]$. This proves the existence of a variable bound as required by (B) of the Preview. Therefore:

Theorem 7.4.9 Ordered resolution with respect to $\prec$ is a decision procedure for finite sets of prefix ordered clauses of basic path logic.

Consequently:

Corollary 7.4.10 Resolution restricted by either $\prec$ or $\prec^\times$ applied to prefix ordered clauses obtained by renaming provides a decision procedure for the satisfiability problem of modal formulae in K, KD and their multi-modal versions.

The obvious question now is, what efficiency gain can be achieved by the ordering refinement and renaming. We assume that renaming is done on any non-literal subformulae and any clause in $S = c^{\Upsilon}(\neg\Pi_f(\varphi^d))$ has the form:

$$E_0[] \vee E_1[\alpha_1] \vee E_2[\alpha_1\alpha_2] \vee \ldots \vee E_n[\alpha_1\alpha_2 \ldots \alpha_n] \vee E_{n+1}[\alpha_1\alpha_2 \ldots \alpha_n u] \tag{7.8}$$

with $0 \leq n < m$ and u either a variable or a constant (Corollary 7.4.4).

Theorem 7.4.11 The form (7.8) is preserved by $\mathcal{R}^{\prec}$- and $\mathcal{R}^{4,\prec^\times}_{N_4}$-resolution with and without condensing.

Proof. Without loss of generality we assume the premises are in normal form. Resolvents of premises that do not include a constant do not include constants either, and have the required form. If a premise includes a constant $\underline{\alpha}$ it occurs in a maximal term with respect to the proper subterm ordering. As the argument of the maximal atom with respect to $\prec$ has as argument a $\lessdot$-maximal term, all variables of the resolvent occur in $A\sigma$ of the ordered resolution rule. The only possible partners for resolution are without constants, or have the same constant $\underline{\alpha}$ in the last position of the $\lessdot$-maximal terms. □

Let **S** be the class of clauses built from a given finite number of predicate and constant symbols with maximal path length m satisfying the following conditions: Any clause is condensed and prefix ordered, and moreover, any clause has the form (7.8).

Theorem 7.4.12 Suppose there is a supply of k predicate symbols, l constant symbols and path lengths do not exceed m.

(i) The number of literals in any clause of **S** is at most $2k(m+1)$.

(ii) The size of **S** is at most $(l+1)2^{2k(m+1)}$.

Proof. (i) The largest clause in the class **S** is

$$\begin{aligned}
& P_1[] \vee \neg P_1[] \vee \ldots \vee P_k[] \vee \neg P_k[] \\
\vee\; & P_1[\alpha_1] \vee \neg P_1[\alpha_1] \vee \ldots \vee P_k[\alpha_1] \vee \neg P_k[\alpha_1] \\
\vee\; & P_1[\alpha_1\alpha_2] \vee \neg P_1[\alpha_1\alpha_2] \vee \ldots \vee P_k[\alpha_1\alpha_2] \vee \neg P_k[\alpha_1\alpha_2] \\
\vee\; & \ldots \\
\vee\; & P_1[\alpha_1 \ldots \alpha_{m-1}] \vee \ldots \vee \neg P_k[\alpha_1 \ldots \alpha_{m-1}] \\
\vee\; & P_1[\alpha_1 \ldots \alpha_{m-1}u] \vee \ldots \vee \neg P_k[\alpha_1 \ldots \alpha_{m-1}u].
\end{aligned}$$

Each line represents $2k$ literals, and in all there are $2k(m+1)$ literals in the clause.

(ii) There are 2^{2k} clauses of the form $E_0[]$, including the empty clause. Because there are $2k\cdot(l+1)$ literals of the form $\pm P_i[u]$, there are $2^{2k\cdot(l+1)}$ non-variant clauses of the form $E_1[u]$. Similarly, we see there are at most $2^{2k\cdot(l+1)}$ non-variant clauses of the form $E_{n+1}[\alpha_1 \ldots \alpha_n u]$ for any $1 \leq n < m$. These are used to build clauses of the form (7.8), and there are then at most

$$2^{2k} \cdot \underbrace{2^{2k} \cdot \ldots \cdot 2^{2k}}_{m \text{ times}} \cdot (l+1)$$

such clauses. There are 2^{2k} choices for $E_0[]$, 2^{2k} choices for any $E_n[\alpha_1 \ldots \alpha_n]$ with $1 \leq n \leq m$, and $l+1$ choices for u in the maximal term. (A more precise bound is $2^{2k} + 2^{2k} \cdot 2^{2k} \cdot (l+1) + \ldots + 2^{2k(m+1)} \cdot (l+1) = 2^{2k}(l+1)(2^{2km}-1)/(2^{2k}-1)$.) □

This gives an indication of the worst case space complexity of ordered resolution for the basic path logic and the associated basic modal logics. Compare this with the bound given in Theorem 6.6.11 for unrestricted resolution and condensing. The complexity for the translation of *S5* is much better since paths have length at most one, namely (i) $4k$ and (ii) $(l+1)2^{4k}$ for the number of literals in any clause and the size of the class **S**, respectively.

7.5 Decidability of transitive modal logics

Unrefined resolution provides decision procedures for transitive modal logics, like *K4* and *S4*, provided an artificial term depth bound is specified in advance (Corollary 6.5.4). This is not very satisfactory. This section discusses the problems of devising an elementary (ordered) resolution decision procedure for path logics including associativity, reviewing the method of Zamov (1989), who claims to have a resolution decision procedure using lock resolution on definitional forms of the path logic associated with *S4*. Unfortunately, due to its brevity and a number of (mostly small) mistakes, Zamov's presentation is extremely difficult reading.

The problem with path logics associated with transitive modal logics is that unification modulo the associativity law expands terms. The language determined by the theories including associativity includes the operator $\circ$ with which complex functional terms can be formed and substituted into variables causing term expansion. Our aim is to find refinements that ensure terms cannot grow indefinitely, for then decidability follows by Theorem 6.5.3. The refinements must achieve two things: control the repetition of constants in paths and control the number of variables in terms.

The following example demonstrates how constants are duplicated during unordered inference. The formula

$$\rho_6 = \Box(\Box(p \to q) \to \neg p) \to \Diamond\neg\Box(p \to q) \tag{7.9}$$

is not a theorem in *KD4*, because $\neg\rho_6 = \Box(\Diamond(p \land \neg q) \lor \neg p) \land \Box\Box(\neg p \lor q)$ is satisfiable in the single world model with both p and q false in the world. The clausal form of ρ_6 is

1. $P[\alpha\underline{\beta}] \lor \neg P[\alpha]$
2. $\neg Q[\alpha\underline{\beta}] \lor \neg P[\alpha]$
3. $\neg P[\gamma\delta] \lor Q[\gamma\delta]$.

The problem is, under associativity the first two clauses produce an infinite set of resolvents.

4. $\neg P[\alpha] \lor \neg Q[\alpha\underline{\beta}\underline{\beta}]$ $\qquad$ $[1, 2, \text{associativity}]$
5. $\neg P[\alpha] \lor \neg Q[\alpha\underline{\beta}\underline{\beta}\underline{\beta}]$ $\qquad$ $[1, 4, \text{associativity}]$

$\vdots$

For *K4*, the solution to the problem of indefinite strings of constants is a suitable form of renaming. Renaming all non-literal subformulae (as defined in the previous section) bears prefix ordered clauses of the form

$$E_0[] \lor E_1[\alpha_1] \lor \ldots \lor E_n[\alpha_1\alpha_2\ldots\alpha_n] \lor E_{n+1}[\alpha_1\alpha_2\ldots\alpha_n u].$$

At most one constant occurs in any such clause (Corollary 7.4.4) and this form is preserved in the ordered calculus under associativity (Theorem 7.4.11), but not in general under identity (recall the example from Section 7.4 on page 139).

We briefly digress to demonstrate what happens for the sample set under an unordered constraint approach as proposed in Mohr (1995). The constraint approach postpones unification to the end of the computation. The input set for ρ_6 reformulated in terms of constraints is

1.	$Px \lor \neg Py$	$\mid\ x \neq [\alpha\underline{\beta}] \lor y \neq [\alpha]$
2.	$\neg Qx \lor \neg Py$	$\mid\ x \neq [\alpha\underline{\beta}] \lor y \neq [\alpha]$
3.	$\neg Px \lor Qy$	$\mid\ x \neq [\gamma\delta] \lor y \neq [\gamma\delta]$.

Resolution steps similar to the above result in increasingly larger constraints:

4. $\neg Py \vee \neg Qx'$	|	$x \neq [\alpha\underline{\beta}] \vee y \neq [\alpha] \vee x' \neq [\alpha'\underline{\beta}] \vee x \neq [\alpha']$	$[1, 2, \text{assoc.}]$
5. $\neg Py \vee \neg Qx''$	|	$x \neq [\alpha\underline{\beta}] \vee y \neq [\alpha] \vee x' \neq [\alpha'\underline{\beta}] \vee x \neq [\alpha']$ $\vee\, x'' \neq [\alpha''\underline{\beta}] \vee x' \neq [\alpha'']$	$[1, 4, \text{assoc.}]$

$\vdots$

Clauses 4. and 5. are condensed. Observe the chains $y, \alpha, x, \alpha', x'$ and $y, \alpha, x, \alpha', x', \alpha'', x''$ of variables across the inequalities. Clearly then, constraints alone do not avoid infinite computation. The problem of indefinite strings of constants for theory resolution is aggravated, because not only do we have repeated occurrences of the same constant (for example $\underline{\beta}$ occurs three times in the constraint of 5.), we note also the number of variables increase rapidly. (End of digression)

The next problem is, how can we control the increase of variables under associativity. The root of the problem is that unification under associativity splits variables and introduces new ones. For example, a most general unified term of $[\alpha\beta\gamma]$ and $[\alpha_1\beta_1\gamma_1]$ is the term $[\gamma_5\gamma_4\gamma_3\gamma_2\gamma_1]$. The reduction of the unification problem for this example is:

$$
\begin{aligned}
&\{[\alpha\beta\gamma] =^? [\alpha_1\beta_1\gamma_1]\} \\
&\quad \rightsquigarrow \quad \{\gamma =^? \gamma_2 \circ \gamma_1,\ \ [\alpha\beta\gamma_2] =^? [\alpha_1\beta_1]\} \\
&\quad \rightsquigarrow \quad \{\gamma =^? \gamma_2 \circ \gamma_1,\ \ \beta_1 =^? \gamma_3 \circ \gamma_2,\ \ [\alpha\beta] =^? [\alpha_1\gamma_3]\} \\
&\quad \rightsquigarrow \quad \{\gamma =^? \gamma_2 \circ \gamma_1,\ \ \beta_1 =^? \gamma_3 \circ \gamma_2,\ \ \beta =^? \gamma_4 \circ \gamma_3,\ \ [\alpha\gamma_4] =^? [\alpha_1]\} \\
&\quad \rightsquigarrow \quad \{\gamma =^? \gamma_2 \circ \gamma_1,\ \ \beta_1 =^? \gamma_3 \circ \gamma_2,\ \ \beta =^? \gamma_4 \circ \gamma_3,\ \ \alpha_1 =^? \gamma_5 \circ \gamma_4,\ \ \alpha =^? \gamma_5\}.
\end{aligned}
$$

The clauses

1. $P[\alpha] \vee Q[\alpha\beta] \vee R'[\alpha\beta\gamma] \vee R[\alpha\beta\gamma]$
2. $Q[\alpha_1] \vee P[\alpha_1\beta_1] \vee R''[\alpha_1\beta_1\gamma_1] \vee \neg R[\alpha\beta\gamma]$

have six resolvents, the one associated with the unifier computed above being:

3. $P[\alpha_1] \vee Q[\alpha_1\alpha_2] \vee Q[\alpha_1\alpha_2\alpha_3] \vee P[\alpha_1\alpha_2\alpha_3\alpha_4]$ $\quad [1, 2, \text{associativity}]$
   $\vee\, R'[\alpha_1\alpha_2\alpha_3\alpha_4\alpha_5] \vee R''[\alpha_1\alpha_2\alpha_3\alpha_4\alpha_5]$.

For *S4* we will now define two simplification rules that reduce the number of variables in clauses. The rules can be viewed as stemming from the reduction law $\Box\Box p \leftrightarrow \Box p$.

Some additional notation will be useful. We introduce notation for dividing and reassembling clauses according to prefixes which is analogous to that for sets of terms used in Chapter 6. First, we extend the definition of the operation of prepending prefixes to clauses. For any clause $C = L_1 t_1 \vee \ldots \vee L_n t_n$, the clause

$$L_1[st_1] \vee \ldots \vee L_n[st_n]$$

obtained from C by adding s as prefix to every term in C will be denoted by $s\,C$. The operation is monotone (if $C \subseteq D$ then $s\,C \subseteq s\,D$).

$C(s)$ denotes the clause obtained from the subclause C' of C of all literals with prefix s by removing the prefix. This operation too, satisfies the same properties as the corresponding operation for sets of terms, namely $(C(s))(t) = C([st])$. For example, let

$$C = Ps \vee Q[s\alpha] \vee R[s\alpha\gamma] \vee Q[s\alpha\beta] \vee R[s\alpha\beta\gamma] \vee P'[s\alpha\beta t]. \tag{7.10}$$

Then

$$\begin{aligned} C(s) &= P[] \vee Q[\alpha] \vee R[\alpha\gamma] \vee Q[\alpha\beta] \vee R[\alpha\beta\gamma] \vee P'[\alpha\beta t] \\ C([s\alpha]) &= Q[] \vee R[\gamma] \vee Q[\beta] \vee R[\beta\gamma] \vee P'[\beta t] \\ C([s\alpha\beta]) &= Q[] \vee R[\gamma] \vee P't. \end{aligned}$$

If $C(s)$ is the clause $L_1t_1 \vee \ldots \vee L_nt_n$ then the subclause of C containing all literals with prefix s is given by $s\,C(s)$ and denotes the largest subclause of the form

$$L_1[st_1] \vee \ldots \vee L_n[st_n].$$

$s\,C(s)$ may be empty and t_i are empty or non-empty strings. For the sample C of (7.10)

$$[s\alpha\beta]C([s\alpha\beta]) = Q[s\alpha\beta] \vee R[s\alpha\beta\gamma] \vee P'[s\alpha\beta t].$$

The first rule we introduce is the following replacement rule:

Replacement of pairs of variables $\quad S \cup \{C = [s\alpha\beta]\ C([s\alpha\beta]) \vee C'\}$
$\quad \triangleright\ S \cup \{D = \ [s\alpha]\ C([s\alpha\beta]) \vee C'\}$

provided the variable α does not occur in C'.

The rule replaces any pair $\alpha\beta$ of variables, which occurs always as a sequence in a variable indecomposable clause, by one variable. Prefix stability ensures that the variable β does not occur in C'. By this rule

$$\begin{aligned} P[\alpha\beta] \vee R[\alpha\beta] &\quad \text{is replaced by} \quad P[\alpha] \vee R[\alpha], \quad \text{and} \\ P[\alpha] \vee R[\alpha\beta\gamma] \vee Q[\alpha\beta\gamma] &\quad \text{is replaced by} \quad P[\alpha] \vee R[\alpha\beta] \vee Q[\alpha\beta]. \end{aligned}$$

Lemma 7.5.1 Let C and D be as in the 'Replacement of pairs of variables' rule. Then C and D are logically equivalent modulo the laws of right identity and associativity.

Proof. It is easy to see that C and D subsume each other, for $C\{\beta \mapsto \underline{e}\} =_{S4} D$ and $D\{\alpha \mapsto \alpha' \circ \beta\} =_{S4} C$. □

Modulo associativity we merely have that D subsumes C, which means that clauses of the form C can be removed from S, when D is in S, and only then.

The second rule removes excess literals from clauses.

Deletion of literals

$$S \cup \{C = [s\alpha\beta]\ C([s\alpha\beta]) \vee L_1[s\alpha t_1] \vee \ldots \vee L_n[s\alpha t_n] \vee C'\}$$
$$\triangleright\ S \cup \{D = \ [s\alpha]\ C([s\alpha\beta]) \vee C'\}$$

provided (i) no term of the form $[s\alpha\ldots]$ occurs in C', (ii) for any $1 \le i \le n$, either $t_i = []$ or $t_i = [ut']$ and $u \ne \beta$, and (iii) the family $\{(L_i, t_i)\}_i$ is a subset of the set $\{(L, t) \mid L[s\alpha\beta t] \in [s\alpha\beta]C([s\alpha\beta])\}$.

The conditions hold for the clause of (7.10) as

$$\begin{aligned}\textstyle\bigvee_i L_i[s\alpha t_i] &= Q[s\alpha] \vee R[s\alpha\underline{\gamma}]\\ [s\alpha\beta]C([s\alpha\beta]) &= Q[s\alpha\beta] \vee R[s\alpha\beta\underline{\gamma}] \vee P'[s\alpha\beta t]\\ C' &= Ps.\end{aligned}$$

Condition (iii) concerns the literal heads and the suffixes of α and β in $\bigvee_i L_i[s\alpha t_i]$ and $[s\alpha\beta]C([s\alpha\beta])$, respectively. The condition is satisfied, because $\{(Q,[]),(R,\underline{\gamma})\}$ is a subset of $\{(Q,[]),(R,\underline{\gamma}),(P',t)\}$. The rule removes $Q[s\alpha] \vee R[s\alpha\underline{\gamma}]$ from C and deletes the variable β from the remainder leaving

$$D = Ps \vee Q[s\alpha] \vee R[s\alpha\underline{\gamma}] \vee P'[s\alpha t].$$

Lemma 7.5.2 C and D as in the 'Deletion of literals' rule are logically equivalent modulo right identity and associativity.

Proof. Assume the set $\{(L_i,t_i)\}_i$ is a subset of the set of pairs (L,t) with $L[s\alpha\beta t] \in [s\alpha\beta]C([s\alpha\beta])$. Then t is empty or a string of constants, for otherwise t has two prefixes, $[s\alpha]$ and $[s\alpha\beta]$, and prefix stability is violated. $\bigvee_i L_i[s\alpha t_i]$ is the subclause

$$[s\alpha](C([s\alpha]) \setminus \beta C([s\alpha\beta]))$$

of C, and neither α nor β occur in C'.

$[s\alpha]C([s\alpha\beta]) \vee C'$ can be obtained from C by instantiating β with $\underline{e}$, normalisation and eliminating $\bigvee_i L_i[s\alpha t_i]$ by standard condensing. This proves the left-to-right direction, that is, C implies D. The right-to-left direction follows since $[s\alpha]C([s\alpha\beta]) \vee C'$ subsumes C, for $N_4([s\alpha]C([s\alpha\beta])\{\alpha \mapsto \alpha' \circ \beta\}) \vee C'$ is a subset of C. □

This result is a generalisation of Zamov's (1989) Lemma 4.5.

Lemma 7.5.3 The two rules are compatible with ordered *S4*-resolution, with and without condensing and normalisation (as defined in Section 7.2).

Proof. For each rule we show, C is redundant in $S \cup \{C, D\}$. Without loss of generality suppose C and D are in normal form. For both rules, $D \prec C$ is a consequence of the property: $[s\alpha t] \prec [s\alpha\beta t]$, for any terms s, t and any variables α, β. (The property is encoded as $f(t, f(\alpha, s)) \prec^f f(t, f(\beta, f(\alpha, s)))$. The relationship is evident since the second argument $f(\alpha, s)$ on the left hand side is a subterm of the second argument $f(\beta, f(\alpha, s))$ of the right hand side.) Then $L[s\alpha t] \prec L[s\alpha\beta t]$ and hence

$$[s\alpha]C([s\alpha\beta]) \vee C' \prec [s\alpha\beta]C([s\alpha\beta]) \vee C'.$$

Also, $[s\alpha]C([s\alpha\beta]) \vee C' \prec [s\alpha\beta]C([s\alpha\beta]) \vee L_1[s\alpha t_1] \vee \ldots \vee L_n[s\alpha t_n] \vee C'$. □

We say two literals L and L' are *adjacent* to each other if the lengths of their arguments differ by one. The 'Deletion of literals' removes repetition in adjacent $L[s\alpha t]$ and $L[s\alpha\beta t]$. For example, one of the Q literals in

$$C = P[\alpha_1] \vee Q[\alpha_1\alpha_2] \vee Q[\alpha_1\alpha_2\alpha_3] \vee P[\alpha_1\alpha_2\alpha_3\alpha_4] \vee R'[\alpha_1\alpha_2\alpha_3\alpha_4\alpha_5]$$

is superfluous. The clause is equivalent to the following:

$$D = P[\alpha_1] \vee Q[\alpha_1\alpha_2] \vee P[\alpha_1\alpha_2\alpha_4] \vee R'[\alpha_1\alpha_2\alpha_4\alpha_5].$$

The two simplification rules do not suffice to avoid duplication of constants and repetition across non-adjacent literals as in

$$P[\alpha_1] \vee Q[\alpha_1\alpha_2] \vee P[\alpha_1\alpha_2\alpha_3] \vee Q[\alpha_1\alpha_2\alpha_3\alpha_4] \vee \ldots .$$

Zamov claims this kind of repetition can also be eliminated, but his argument is not conclusive. Also, I see a problem with eliminating duplication in certain cases of adjacent $E_i[\ldots]$, like for instance in the clause

$$C = R'[\alpha_1] \vee P[\alpha_1\alpha_2] \vee Q[\alpha_1\alpha_2] \vee Q[\alpha_1\alpha_2\alpha_3] \vee P'[\alpha_1\alpha_2\alpha_3] \vee R''[\alpha_1\alpha_2\alpha_3\alpha_4]$$

where $\{(P,[]),(Q,[])\}$ is not a subset of $\{(Q,[])\}$ (as required by (iii) of the 'Deletion of literals' rule).

Now, we consider an example which I chose because the proof by unrestricted resolution duplicates constants, and analogously the proof by the semantic diagram method includes a label with duplicate constants. The formula

$$\rho_7 = \Diamond\underset{\underline{\beta}}{\Box}(\underset{\underline{\gamma}}{\Diamond}\Box p \to \underset{\underline{\delta}}{\Box} p) \tag{7.11}$$

is a theorem in *S4* (and also in *KD4*). A clausal form obtained by transformation to negation normal form and Skolemisation is

1. $P[\alpha\underline{\beta}\underline{\gamma}\delta]$
2. $\neg P[\alpha\underline{\beta}\underline{\delta}]$.

It reduces to the empty clause with the most general unifiers of $\{[\alpha\underline{\beta}\underline{\gamma}\delta] =^? [\alpha'\underline{\beta}\underline{\delta}]\}$ being:

$$\sigma_1 = \{\alpha' \mapsto \alpha \circ \underline{\beta} \circ \underline{\gamma} \circ \delta',\ \ \delta \mapsto \delta' \circ \underline{\beta} \circ \underline{\delta}\}$$
$$\sigma_2 = \{\alpha' \mapsto \alpha \circ \underline{\beta} \circ \underline{\gamma},\ \ \delta \mapsto \underline{\beta} \circ \underline{\delta}\}.$$

Then $[\alpha\underline{\beta}\underline{\gamma}\delta]\sigma_1 =_4 [\alpha\underline{\beta}\underline{\gamma}\delta'\underline{\beta}\underline{\delta}]$ and $[\alpha\underline{\beta}\underline{\gamma}\delta]\sigma_2 =_4 [\alpha\underline{\beta}\underline{\gamma}\underline{\beta}\underline{\delta}]$. Note the duplicate occurrence of the constant symbol $\underline{\beta}$.

A proof using the semantic diagram method of Hughes and Cresswell (1996) is presented in Figure 7.1. The values 1 and 0 below a symbol are the assignments (true or false) to the corresponding subformulae. To begin with, in the initial world [] the formula ρ_7 is assigned the value 0 (thus, the 0 below the $\Diamond$ in the first line). Assignments for modal subformulae are also marked by an asterisk either in the superscript or subscript. Asterisks have the following meaning: If $\Diamond\psi$ is false in a world x then ψ is false in all accessible worlds, which is indicated by the $*$ in the superscript of 0^*. ψ with the assignment false will be carried over to all subsequent worlds. We assume reflexivity, so, ψ must be false in x, which explains the 0 below the $\Box$. If $\Box\psi$ is assigned false in the world x then ψ is false in some successor word. This information, that is, the existence of a world in which ψ is false, is carried by the subscript of 0_* below the $\Box$. Accordingly, we create a successor world which we name $[\underline{\beta}]$ to illustrate the analogy to the resolution proof, and so forth. The bold face assignments in the last line are contradictory, which means we have found a refutation.

$$
\begin{array}{rl}
[\,] & \Diamond\,\Box\,(\Diamond\Box p \rightarrow \Box p) \\
 & 0^{*}0_{*} \\
[\underline{\beta}] & \Diamond\Box p \rightarrow\ \Box p,\quad \Box\,(\Diamond\Box p \rightarrow \Box p) \\
 & 1_{*} \qquad 0\ \ 0_{*} \qquad 0_{*} \\
[\underline{\beta}\underline{\gamma}] & \Box p,\quad \Box\,(\Diamond\Box p \rightarrow \Box p) \\
 & 1^{*}1\quad 0_{*}\ \ 1\,1\,1\ \ 1\ \ 1\,1 \\
[\underline{\beta}\underline{\gamma}\underline{\beta}] & \Diamond\Box p \rightarrow\ \Box p,\quad p,\quad \Box\,(\Diamond\Box p \rightarrow \Box p) \\
 & 1^{*} \qquad 0\ \ 0_{*}\,0\quad 1\quad 0_{*}\ \ 1\,1\,1\ \ 1\ \ 1\,1 \\
[\underline{\beta}\underline{\gamma}\underline{\beta}\underline{\delta}] & p,\quad p,\quad \Box\,(\Diamond\Box p \rightarrow \Box p) \\
 & \mathbf{0}\quad \mathbf{1}\quad 0_{*}\ \ 1\,1\,1\ \ 1\ \ 1\,1
\end{array}
$$

Figure 7.1: A tableaux refutation of (7.11)

An ordered resolution proof exists without any occurrences of constants in resolvents. The ingredients are renaming of all non-literal subformulae and a particular ordering of the predicate constants. Relying on Theorem 7.3.2 for

$$
\begin{aligned}
\neg\rho_7^d = {} & \Box(a_1 \rightarrow \Box p) \wedge \Box(a_2 \rightarrow \Diamond a_1) \wedge \Box(a_3 \leftarrow \Box p) \wedge \Box(a_4 \leftarrow (a_2 \rightarrow a_3)) \wedge \\
& \Box(a_5 \leftarrow \Box a_4) \wedge \Box(a_6 \leftarrow \Diamond a_5) \wedge \neg a_6,
\end{aligned}
$$

and the fact that *S4* is a serial modal logic, the clausal form $c^{\Upsilon}(\neg\Pi_f(\rho_7^d))$ is given by clauses 1. to 8. of Figure 7.2. The figure presents a derivation of the empty clause. The black dots mark the clauses which constitute a proof and indicate how often clauses are used. The applied ordering is determined by the following ordering on the predicate symbols

$$A_6 \prec A_5 \prec A_4 \prec A_2 \prec A_1 \prec A_3 \prec P.$$

This ordering was chosen in accordance with Zamov's specification for his lock resolution procedure. He requires that (i) complementary literals have identical values, (ii) literals $L_1 s$ with an argument that is a prefix of the argument of a literal $L_2[st]$, has a smaller value than $L_2[st]$, and (iii) literals complementary to a literal with an occurrence of a constant have smaller values than literals complementary to a literal with no occurrence of a constant.

Zamov relies on being able to dispense with clauses that contain constants, in his Lemma 4.1. The statement of the lemma seems to claim for a refutable set S originating from a modal theorem there is a resolution proof that does not include constants except for those occurring in the initial clauses of S. This cannot be, because under lock resolution on maximal literals, the literal $A_4[\alpha\underline{\beta}]$ in clause 11. of Figure 7.2 is not maximal and the derivation of the empty clause without using resolvents containing constants is impossible. But, the proof of the lemma seems to be concerned with a different problem altogether. It argues resolvents of any premises in S do not include constants. This cannot be, either. A counter example is the clause 9. Possibly it was overlooked that $\{\delta \mapsto \delta' \circ \underline{\beta},\ \ \alpha \mapsto \gamma \circ \delta'\}$ is also a most general unifier of the problem $[\alpha\underline{\beta}] =^{?} [\gamma\delta]$.

For the moment the only solution for deciding satisfiability in path logics associated with transitive modal logics is to use imparted term depth bounds. Another solution that will undoubtedly work, is to simulate the corresponding tableaux decision algorithms by resolution.

•	1.	$\neg A_1[\alpha] \vee \underline{P[\alpha\beta]}$		
•	2.	$\neg A_2[\alpha] \vee \underline{A_1[\alpha\underline{\gamma}]}$		
•	3.	$A_3[\alpha] \vee \underline{\neg P[\alpha\underline{\delta}]}$		
•	4.	$A_4[\alpha] \vee \underline{A_2[\alpha]}$		
•	5.	$A_4[\alpha] \vee \underline{\neg A_3[\alpha]}$		
••	6.	$A_5[\alpha] \vee \underline{\neg A_4[\alpha\underline{\beta}]}$		
••	7.	$A_6[\alpha] \vee \underline{\neg A_5[\alpha\beta]}$		
•	8.	$\underline{\neg A_6[]}$		
	9.	$\underline{\neg A_1[\alpha\underline{\gamma}]} \vee A_3[\alpha]$	$[1, 3, \text{id.}, \text{assoc.}, N_{T4}]$	
•	10.	$\neg A_1[\alpha] \vee \underline{A_3[\alpha\beta]}$	$[1, 3, \text{id.}, \text{assoc.}, N_{T4}]$	
•	11.	$\underline{A_4[\alpha\beta]} \vee \underline{\neg A_1[\alpha]}$	$[5, 10, \text{id.}, \text{assoc.}, N_{T4}]$	
	12.	$\neg A_2[\alpha] \vee \underline{A_4[\alpha\underline{\gamma}\beta]}$	$[2, 11, \text{id.}, \text{assoc.}, N_{T4}]$	
•	13.	$\underline{A_5[\gamma\delta']} \vee \underline{\neg A_1[\gamma]}$	$[6, 11, \text{id.}, \text{assoc.}, N_{T4}]$	
	14.	$A_5[\alpha] \vee \underline{\neg A_1[\alpha\underline{\beta}]}$	$[6, 11, \text{id.}, \text{assoc.}, N_{T4}]$	
	15.	$\neg A_2[\alpha] \vee \underline{A_5[\alpha\underline{\gamma}\beta]}$	$[6, 12, \text{id.}, \text{assoc.}, N_{T4}]$	
	16.	$\neg A_2[\alpha] \vee \underline{A_6[\alpha\underline{\gamma}\beta]}$	$[7, 15, \text{id.}, \text{assoc.}, N_{T4}]$	
	17.	$\neg A_2[\alpha] \vee \underline{A_5[\alpha\underline{\gamma}\delta]}$	$[2, 13, \text{id.}, \text{assoc.}, N_{T4}]$	(variant of 15.)
	18.	$\underline{A_6[\alpha\beta]} \vee \underline{\neg A_1[\alpha]}$	$[7, 13, \text{id.}, \text{assoc.}, N_{T4}]$	
•	19.	$A_6[\alpha] \vee \underline{\neg A_1[\alpha\beta]}$	$[7, 13, \text{id.}, \text{assoc.}, N_{T4}]$	
	20.	$\neg \underline{A_1[]}$	$[8, 18, \text{id.}, \text{assoc.}, N_{T4}]$	
	21.	$\neg A_2[\alpha] \vee \underline{A_6[\alpha\underline{\gamma}\beta]}$	$[2, 18, \text{id.}, \text{assoc.}, N_{T4}]$	(variant of 16.)
•	22.	$\underline{\neg A_2[\alpha\beta]} \vee A_6[\alpha]$	$[2, 19, \text{id.}, \text{assoc.}, N_{T4}]$	
	23.	$\neg A_2[\alpha] \vee \underline{A_6[\alpha\underline{\gamma}]}$	$[2, 19, \text{id.}, \text{assoc.}, N_{T4}]$	(subsumed by 16. & 21.)
•	24.	$\underline{A_4[\alpha\beta]} \vee A_6[\alpha]$	$[4, 22, \text{id.}, \text{assoc.}, N_{T4}]$	
•	25.	$\underline{A_5[\alpha\beta]} \vee A_6[\alpha]$	$[6, 24, \text{id.}, \text{assoc.}, N_{T4}]$	
	26.	$A_5[\alpha] \vee \underline{A_6[\alpha\underline{\beta}]}$	$[6, 24, \text{id.}, \text{assoc.}, N_{T4}]$	
•	27.	$\underline{A_6[\alpha\beta]} \vee A_6[\alpha]$	$[7, 25, \text{id.}, \text{assoc.}, N_{T4}]$	
	28.	$A_6[\alpha] \vee \underline{A_6[\alpha\beta]}$	$[7, 25, \text{id.}, \text{assoc.}, N_{T4}]$	(variant of 27.)
•	29.	$\underline{A_6[\alpha]}$	$[7, 25, \text{repl. of pairs of vars}]$	
•	30.	$\emptyset$	$[8, 29, \text{id.}, \text{assoc.}, N_{T4}]$	

Figure 7.2: A refutation of (7.11) by $\mathcal{R}^{S4,\prec^{\times}}_{\text{COND} \,\circ\, N_{T4}}$-resolution.

7.6 Conclusion

The previous chapter showed for many path logics and the corresponding modal logics unrestricted (theory) resolution is a decision procedure. The search space is enormous, as the largest possible non-redundant clause in the search space has m-story exponential size. In this chapter we studied ordered E-resolution and the method of renaming. Renaming provides an efficient means of transformation to clausal form. We gave a completeness proof for saturation up to redundancy by ordered E-resolution. We looked at finitely based theories which form convergent rewrite systems. The ordered calculus is restricted by a lexicographic combination $\prec^{\times}$ determined by the ordering $\prec$ that orients the rewrite rules of E and the ordering on the unique normal forms. Decidability for basic path logic follows easily when we employ renaming (rendering prefix ordered clauses). An indication of the gain in efficiency for the basic path logic (with empty E) is the single story exponential space requirement in the worst case.

By the general decidability result of the previous chapter, ordered E-resolution with respect to $\prec^{\times}$ is a decision procedure for path logic associated with K, KD, KT, $KD4$ and $S4$, for the latter with a given term depth bound. A remaining challenge is to develop decision procedures not relying on given bounds.

Finally, we note that E-unification problems for prefix ordered clauses often have simpler solutions. In general, when using a calculus requiring semantic factoring we would need general E-unification. However, it turns out that for assciativity alone (that is $KD4$) the unification algorithm of singleton problems (of Section 5.7) suffices. This is not the case for KT or $S4$. For instance, $P[\alpha\beta] \vee P[\alpha\beta\gamma]$ has no (most general) $KD4$-unifier, nor has it a syntactic (most general) unifier. But, the clause has a most general KT and $S4$-unifier, namely $\{\gamma \mapsto \underline{e}\}$. I envisage that in these cases simple add-ons to the algorithms of Section 5.7 will be more efficient than general E-unification.

Chapter 8

Conclusion

As outlined in the Introduction the goal has been to develop optimised resolution based inference methods for propositional modal logics. To this end, I studied the functional translation approach and its optimisation. I described how the relational semantics systematically transforms to the functional alternative semantics on which the corresponding functional translation is based. As discussed in Chapter 2, the fundamental idea is that of decomposing any binary relation into a set of functions and a set identifying the dead-ends of the relation. This bears a translation of any modal formula into a decidable monadic first-order fragment. It involves a special function symbol for encoding accessibility in terms of paths. I named this logic non-optimised path logic and presented a new direct definition independent of modal logic. I noted that any non-serial modal logic can embedded in a serial modal logic by adjoining a special variable for recording undefinability.

The next transformation employs the operation by which existential and universal quantifiers are swapped. Semantically, this optimisation is embodied in special maximal or patched functional models, which are important for another reason. Namely, certain functional correspondence properties can be simplified by eliminating the dependency of the functional quantifiers on the world quantifier. I exhibit in Chapter 3 the optimised translation applies not only to formulae we aim to refute, it applies also to axiom schemas and produces approximate first-order formulations for modal schemas that are not equivalently first-order definable. Consequently, first-order inference techniques become available for genuine second-order logics, like K extended with McKinsey's schema and the modal logic $\overline{K}_E$.

The logic $\overline{K}_E$ is a new multi-modal logic which approximates the graded modal logic $\overline{K}$. It serves as an intermediary logic for realising reasoning about numerical quantifiers in a first-order resolution calculus enhanced with simple arithmetic, as described in Chapter 4. The chapter is a fairly complex application of the translation techniques and illustrates the power of our approach.

The optimised functional translation associates propositional modal logics with a lattice of clausal logics, called path logics, defined in Chapter 5. The weakest path logic associated with K, KD and $K_{(m)}$ and its serial extension is the basic path logic. Path logics have a number of pleasant properties important for the decidability result. Input clauses of path logic contain no Skolem function symbols other than constant symbols, which avoids growth of terms during unification. The second essential property is prefix stability of the terms, which remains invariant under deduction. Chapter 5 studies E-unification for T and 4 in greater detail. It presents an improved algorithm for T and 4 that is based on mutation rules, which have the advantage that terms are worked off top down making paramodulation into terms superfluous.

Path logics are of general interest in the fields of logic, computational complexity and resolution theorem proving. In particular, the basic path logic is a new solvable class that is PSPACE-complete (because K, KD and $K_{(m)}$ are). The basic path logic is in fact isomorphic to a reduct of the Bernays-Schönfinkel class (a solvable class of $\exists^*\forall^*$ prefix formulae). Chapter 6 is devoted to the problem of deciding satisfiability in basic path logic by resolution and condensing. As far as I know, basic path logic is the only non-trivial solvable class for which no additional refinements are needed. Moreover, in contrast to many decidability proofs for other solvable classes, the increase of the functional nesting of terms is not a problem in basic path logic. Terms do not expand during deduction. However, the numbers of literals and variables in clauses may increase. Our proof of the existence of literal and variable bounds relies on a new technique of nested prefix partitioning of any set of terms. A complex encoding of variables that accommodates condensedness by the antichain property in a finite power set algebra allows us to give a worst case limit of the size of generated clauses. For a number of path logics and their modal counterparts, namely $S5$ and $K_{(m)}$ and its extensions with D, T and 4 modalities, decidability by resolution combined with condensing and any compatible refinements follows (and possibly a given term depth bound).

Modern theorem provers use forms of ordering restrictions and sophisticated renaming techniques. Chapter 7 defines a general ordered E-resolution calculus based on saturation up to redundancy and studies the clausal forms produced by renaming on the modal level. There is a marked increase in efficiency quantified by the worst case single story exponential space requirement as opposed to the m-story exponential space requirement, for basic path logic. I analysed the problem of deciding the path logics associated with transitive modal logics, which still needs resolving.

Based on the results obtained, I claim the endeavour of doing modal inference by the optimised functional translation into first-order logic and conventional resolution theorem proving is a promising and viable alternative approach. The following explains the advantages and discusses some open problems.

Automatic transformation: The basic procedure and the basic framework for different modal logics is always the same. The transformation of modal formulae to clausal form can be done automatically with linear time and space overhead. For the most, the transformation of axiomatisations to path logics can be done automatically by using the Gabbay-Ohlbach method. Finding appropriate and efficient reasoning mechanisms for testing satisfiability with respect to theories, especially with respect to theories not treated in this thesis, deserves further investigation.

Expressive power: The optimised functional translation is more powerful than the relational translation. For one, it captures also essentially second-order modal logics in a first-order setting. Two, the optimised functional language is more fine-grained than the relational language. Paths carry more information, in a concise manner. In particular, paths and prefixes make the history of worlds explicit, which relational expressions do not do directly. It so happens that the point that paths embody information about predecessor worlds can also be exploited in tableaux approaches as is done in the different prefixed tableaux calculi of Fitting (1983), Massacci (1994), Governatori (1995) and others (see Section 6 in Goré 1995).

Decidability: I have presented elementary resolution procedures that decide satisfiability for non-transitive modal logics. No additional refinement restrictions are required to guarantee termination, with practical implications especially for casual users. Users wanting to test the satisfiability or theoremhood of a modal formula may use his or her favourite resolution theorem prover (provided it is fair). I covered only a selected number of modal logics and the corresponding path theories. Accordingly, much remains to be done. Some open questions are: Exactly which modal logics and path logics have the three properties required by the general decidability result? For given path theories, are there terminating and efficient unification procedures? It is open whether there is or can be a uniform approach bearing terminating algorithms for a larger subclass of path logics. Can the decidability results be generalised to decidable path logics that do not have the particular normal forms we require, for example the path logic associated with the modal logic KM? A particularly important problem concerns the practical solution of decidability by resolution for transitive modal logics. Connected with this, how can periodicity be detected in a resolution calculus? Does there exist a modal logic without the finite model property and a corresponding path logic decidable by a resolution procedure?

Possible extensions: Decidability of modal theoremhood by translation to first-order logic circumvents the limitations of special purpose modal reasoners regarding extendibility. Within the framework of first-order logic and resolution theorem proving there is much room for extension and further exploration. It is conceivable that extending the translation technique is easier in many cases than extending both the theory and the implementation of a tableaux or sequent theorem prover. Extensions that come to mind are relational operations as in propositional dynamic logic and the different KL-ONE variants above $\mathcal{ALC}$, extensions with multi-dimensional modal operators like Tarskian set constraint operators (excluding the μ-operator) and modal predicate logics. It is very likely that there are interesting and useful combinations of propositional modal logics with decidable fragments of first-order logic that can be accommodated in the functional framework. Reasoning with nominals, known as ABox elements in the description logic context, poses no serious problem, even though this means reasoning with equations. I envisage a superposition calculus restricted by the ordering we used in Chapter 7 and conjecture it provides a decision procedure.

Availability of sophisticated theorem provers: Sophisticated and well-tested resolution theorem provers are freely available. OTTER is the most widely used and best known first-order theorem prover (McCune 1994). Two theorem provers which are based on the resolution and superposition framework of Bachmair and Ganzinger (1994, 1997) are SPASS (Weidenbach et al. 1996, Weidenbach 1997) and Saturate (Nivela and Nieuwenhuis 1993). What we seem to lack at this moment are theorem provers for theory resolution, but a feasible, though not optimal, alternative is to use superposition and paramodulation instead.

Practical experience: We have some practical experience, and very encouraging benchmark results are available with which I want to conclude.

The decision result of this thesis initiated an empirical investigation by U. Hustadt of available implementations of decision procedures for $K_{(m)}$ including our approach combining the optimised functional translation and first-order resolution. The tests were done on a large set of randomly generated multi-modal formulae. The graphs of Figures 8.1 and 8.2 are an extract from an extensive analysis reported on in Hustadt and Schmidt (1997a, 1997b, 1997c), and Hustadt et al. (1998).

The systems evaluated are the $\mathcal{KRIS}$ system developed for description logics including $\mathcal{ALC}$ which uses a tableaux method (Baader and Hollunder 1991), a system called KSAT based on a Davis-Putnam algorithm for propositional modal logic (Giunchiglia and Sebastiani 1996a), the Logics Workbench (LWB) which is based on a sequent calculus (Heuerding and Schwendimann 1996), OTTER Version 3.0 with the hyper-resolution setting (McCune 1994), and the prototype SPASS Version 0.42 implementing standard resolution restricted by an extension of the Knuth-Bendix ordering (Weidenbach et al. 1996, Weidenbach 1997). The graphs are percentile graphs. Each point in any graph is a triple (t, p, c) which carries the information that, p out of one hundred randomly generated formulae with complexity c were each solved in at most t seconds. The complexity of the formulae is determined by the ratio of the number of conjunctions over the number of propositional variables. The fixed parameters for the generated formulae are: five propositional variables, three disjuncts in any clause and modal degree two for the graphs of Figure 8.1 and four propositional variables, three disjuncts in any clause and modal degree one for the graphs of Figure 8.2.

The sets of formulae for the first figure were generated according to the scheme of Giunchiglia and Sebastiani (1996a, 1996b). This scheme has some shortcomings that is conveyed by the valleys on the right for all theorem provers except for SPASS. For the concerned region very few hard formulae are generated. Harder formulae are obtained by filtering out certain trivially unsatisfiable formulae.[1] The performance observed, then, is as depicted in the second figure. The benchmarks demonstrate, in practice, performance depends very much on the refinements used and the sophistication of the implementation. Also, they indicate the competitiveness of utilising resolution theorem provers for modal theorem proving.

[1] If a generated formula has a subset of purely propositional clauses that is unsatisfiable then the formula is said to be trivially unsatisfiable.

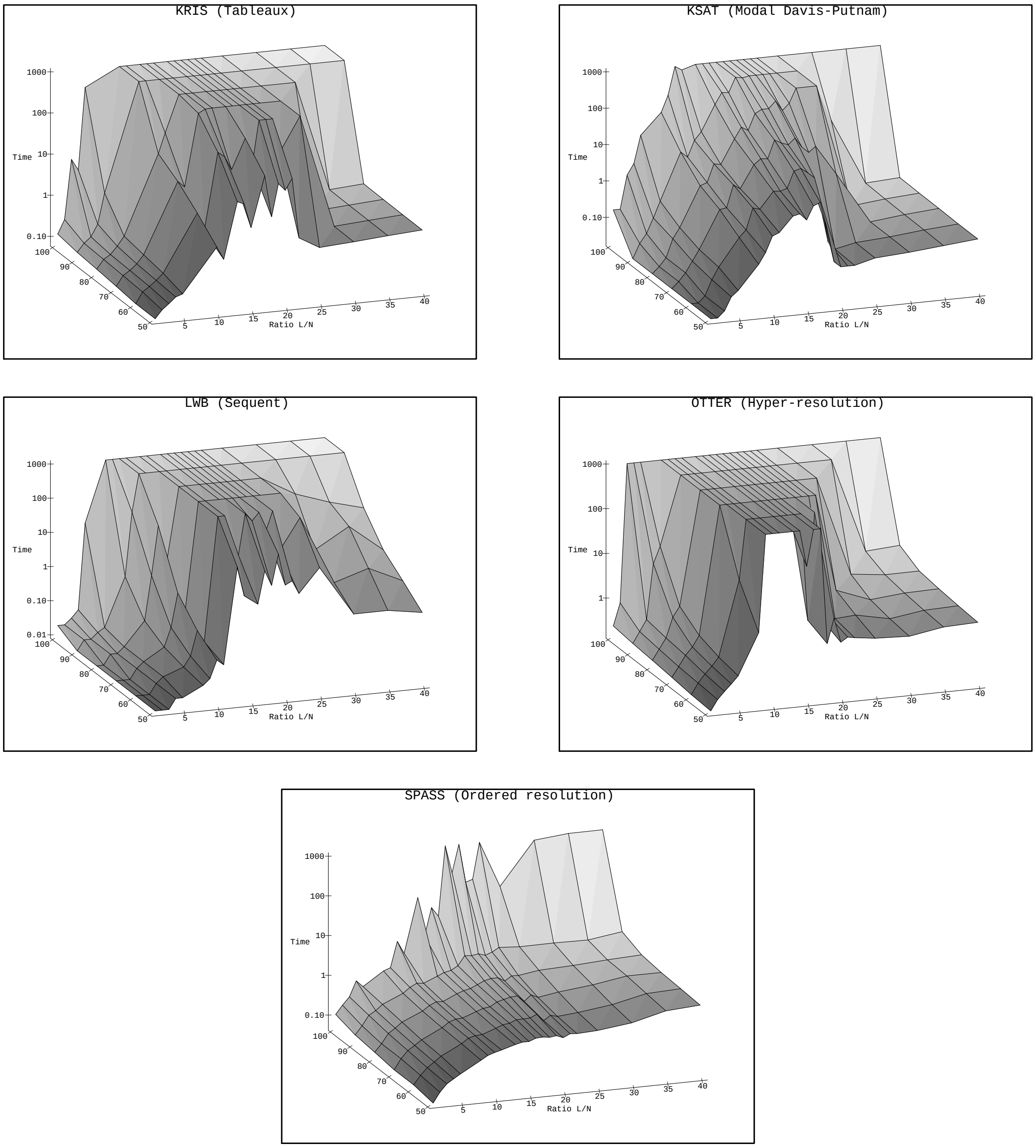


Figure 8.1: The performance of different theorem provers for $K_{(m)}$

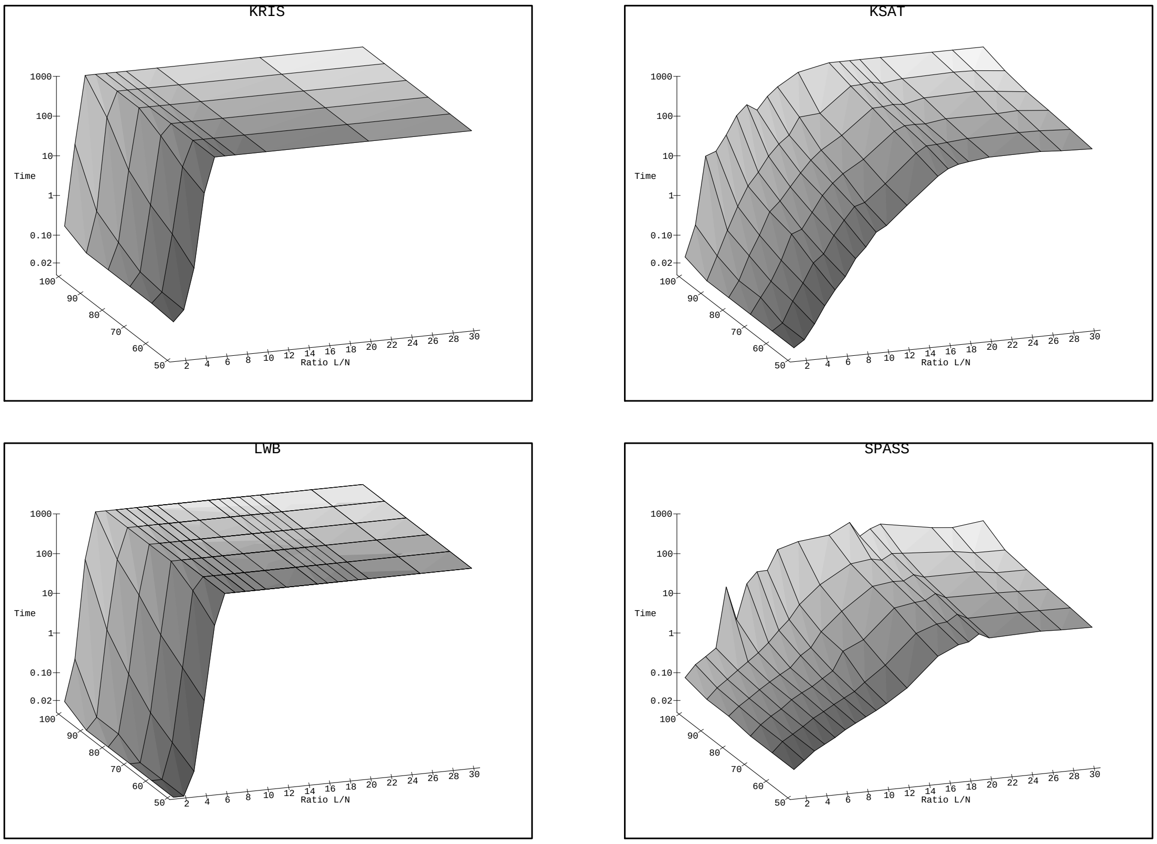


Figure 8.2: The performance for a harder problem set

Bibliography

Anderson, R. and Bledsoe, W. W. (1970), A linear format for resolution with merging and a new technique for establishing completeness, *J. ACM* **17**, 525–534.

Andréka, H., Németi, I. and Sain, I. (1995), On interpolation, amalgamation, universal algebra and Boolean algebras with operators, Unpublished manuscript, Univ. Budapest.

Arnborg, S. and Tidén, E. (1985), Unification problems with one-sided distributivity, *in* J.-P. Jouannaud (ed.), *Proc. Intern. Conf. on Rewriting Techniques and Applications*, Vol. 202 of *Lecture Notes in Computer Science*, Springer, pp. 398–406.

Auffray, Y. and Enjalbert, P. (1992), Modal theorem proving: An equational viewpoint, *J. Logic Computat.* **2**(3), 247–297.

Baader, F. and Hollunder, B. (1991), $\mathcal{KRIS}$: $\mathcal{K}$nowledge $\mathcal{R}$epresentation and $\mathcal{I}$nference $\mathcal{S}$ystem: System description, *Technical Memo TM-90-03*, DFKI, Kaiserslautern.

Baader, F. and Siekmann, J. H. (1993), Unification theory, *in* D. M. Gabbay, C. J. Hogger and J. A. Robinson (eds), *Handbook of Logic in Artificial Intelligence and Logic Programming*, Vol. 2, Oxford Science Publ., pp. 41–125.

Bachmair, L. and Ganzinger, H. (1994), Rewrite-based equational theorem proving with selection and simplification, *J. Logic Computat.* **4**(3), 217–247.

Bachmair, L. and Ganzinger, H. (1997), A theory of resolution, *Research Report MPI-I-97-2-005*, Max-Planck-Institut für Informatik, Saarbrücken. To appear in *Handbook of Automated Reasoning*.

Bachmair, L., Ganzinger, H. and Waldmann, U. (1993), Superposition with simplification as a decision procedure for the monadic class with equality, *in* G. Gottlob, A. Leitsch and D. Mundici (eds), *Proceedings of the Third Kurt Gödel Colloquium (KGC'93)*, Vol. 713 of *Lecture Notes in Computer Science*, Springer, pp. 83–96. The long version is Research Report MPI-I-93-204, Max-Planck-Institut für Informatik, Saarbrücken.

Baumgartner, P. (1992), An ordered theory resolution calculus, *in* A. Voronkov (ed.), *Proc. LPAR'92*, Vol. 624 of *Lecture Notes In Artificial Intelligence*, Springer, pp. 119–130.

Boy de la Tour, T. (1992), An optimality result for clause form translation, *J. Symbolic Computat.* **14**, 283–301.

Bull, R. A. and Segerberg, K. (1984), Basic modal logic, *in* D. Gabbay and F. Guenther (eds), *Handbook of Philosophical Logic*, Vol. II, Reidel, pp. 1–88.

Cerrato, C. (1990), General canonical models for graded normal logics (Graded modalities IV), *Studia Logica* **49**, 241–252.

Chellas, B. F. (1980), *Modal Logic: An Introduction*, Cambridge Univ. Press.

Comon, H., Haberstrau, M. and Jouannaud, J.-P. (1994), Syntacticness, cycle-syntacticness and shallow theories, *Inform. and Computat.* **111**(1), 154–191.

de Caro, F. (1988), Graded modalities II, *Studia Logica* **47**, 1–10.

Dershowitz, N. and Jouannaud, J.-P. (1990), Rewrite systems, *in* J. van Leeuwen (ed.), *Handbook of Theoretical Computer Science B: Formal Methods and Semantics*, North-Holland, chapter 6, pp. 243–309.

Doggaz, N. and Kirchner, C. (1991), Completion for unification, *Theoret. Computer Sci.* **85**, 231–251.

Eder, E. (1992), *Relative Complexities of First Order Calculi*, Artificial Intelligence, Vieweg, Wiesbaden.

Eisinger, N. and Ohlbach, H. J. (1993), Deduction systems based on resolution, *in* D. M. Gabbay, C. J. Hogger, J. A. Robinson and J. Siekmann (eds), *Handbook of Logic in Artificial Intelligence and Logic Programming: Logical Foundations*, Vol. 1, Clarendon Press, Oxford, pp. 183–271.

Fariñas del Cerro, L. and Herzig, A. (1988), Linear modal deductions, *in* E. Lusk and R. Overbeek (eds), *Proc. CADE'88*, Vol. 310 of *Lecture Notes in Computer Science*, Springer, pp. 487–499.

Fariñas del Cerro, L. and Herzig, A. (1989), Automated quantified modal logic, *in* P. B. Brazdil and K. Konolige (eds), *Machine Learning, Meta-Reasoning and Logics*, The Kluwer International Series in Engineering and Computer Science: Knowledge Representation, Learning and Expert Systems, Kluwer, pp. 301–317.

Fariñas del Cerro, L. and Herzig, A. (1995), Modal deduction with applications in epistemic and temporal logics, *in* D. M. Gabbay, C. J. Hogger and J. A. Robinson (eds), *Handbook of Logic in Artificial Intelligence and Logic Programming: Epistemic and Temporal Reasoning*, Vol. 4, Clarendon Press, Oxford, pp. 499–594.

Fattorosi-Barnaba, M. and Cerrato, C. (1988), Graded modalities III, *Studia Logica* **47**, 99–110.

Fattorosi-Barnaba, M. and de Caro, F. (1985), Graded modalities I, *Studia Logica* **44**, 197–221.

Fermüller, C., Leitsch, A., Tammet, T. and Zamov, N. (1993), *Resolution Method for the Decicion Problem*, Vol. 679 of *Lecture Notes in Computer Science*, Springer.

Fine, K. (1969), *For Some Propositions and so Many Possible Worlds*, PhD thesis, Univ. Warwick.

Fine, K. (1972), In so many possible worlds, *Notre Dame J. Formal Logic* **13**(4), 516–520.

Fine, K. (1975), Normal forms in modal logic, *Notre Dame J. Formal Logic* **16**, 229–237.

Fitting, M. (1983), *Prefixed Tableau Systems*, Vol. 169 of *Synthese Library: Studies in Epistemology, Logic, Methodology, and Philosophy of Science*, Reidel, chapter 8.

Fitting, M. (1993), Basic modal logic, *in* D. M. Gabbay, C. J. Hogger, J. A. Robinson and J. Siekmann (eds), *Handbook of Logic in Artificial Intelligence and Logic Programming: Logical Foundations*, Vol. 1, Clarendon Press, Oxford, pp. 365–448.

Gabbay, D. M. (1981), An irreflexivity lemma with applications to axiomatizations of conditions on linear frames, *in* U. Mönnich (ed.), *Aspects of Philosophical Logic*, Reidel, pp. 67–89.

Gabbay, D. M. and Ohlbach, H. J. (1992), Quantifier elimination in second-order predicate logic, *S. Afr. Computer J.* **7**, 35–43. Also in *Proc. KR'92*, 425–436.

Gallier, J. and Snyder, W. (1992), Designing unification procedures using transformations: A survey, *in* Y. N. Moschovakis (ed.), *Logic From Computer Science (Berkeley, CA, 1989)*, Vol. 21 of *Math. Sci. Res. Inst. Publ.*, Springer, New York, pp. 153–215.

Gasquet, O. (1994), *Déduction automatique en logique multi-modale par traduction*, PhD thesis, Univ. Paul-Sabatier, Toulouse.

Giunchiglia, F. and Sebastiani, R. (1996a), Building decision procedures for modal logics from propositional decision procedures: The case study of modal K, *in* M. A. McRobbie and J. K. Slaney (eds), *Automated Deduction: CADE-13*, Vol. 1104 of *Lecture Notes in Artificial Intelligence*, Springer, pp. 583–597.

Giunchiglia, F. and Sebastiani, R. (1996b), A SAT-based decision procedure for $\mathcal{ALC}$, *in* L. C. Aiello, J. Doyle and S. Shapiro (eds), *Proc. KR'96*, Morgan-Kaufmann, pp. 304–314.

Goble, L. F. (1970), Grades of modality, *Logique et Analyse* **13**, 323–334.

Goldblatt, R. (1987), *Logics of Time and Computation*, Vol. 7 of *CSLI Lecture Notes*, Chicago Univ. Press, Chicago.

Goré, R. (1995), Tableau methods for modal and temporal logics, *Technical Report TR-ARP-15-95*, Australian National Univ., Canberra. To appear in D'Agostino, M., Gabbay, D., Hähnle, R. and Posegga, J. (eds), *Handbook of Tableau Methods*, Kluwer.

Gottlob, G. and Fermüller, C. G. (1993), Removing redundancy from a clause, *Artificial Intelligence* **61**, 263–289.

Governatori, G. (1995), Labelled tableaux for multi-modal logics, *in* P. Baumgartner, R. Hähnle and J. Posegga (eds), *Theorem Proving with Analytic Tableaux and Related Methods*, Vol. 918 of *Lecture Notes in Artificial Intelligence*, Springer, pp. 79–94.

Herzig, A. (1989), *Raisonnement automatique en logique modale et algorithmes d'unification.*, PhD thesis, Univ. Paul-Sabatier, Toulouse.

Herzig, A. (1990), A new decidable fragment of first order logic. In *Abstracts of the 3rd Logical Biennal, Summer School and Conference in Honour of S. C. Kleene*, Varna, Bulgaria.

Heuerding, A. and Schwendimann, S. (1996), On the modal logic K plus theories, *in* H. Kleine Büning (ed.), *Proc. CSL'95*, Vol. 1092 of *Lecture Notes in Computer Science*, Springer, pp. 308–319.

Hughes, G. E. and Cresswell, M. J. (1984), *A Companion to Modal Logic*, Methuen, London.

Hughes, G. E. and Cresswell, M. J. (1996), *A New Introduction to Modal Logic*, Routledge, London.

Hustadt, U. (1997), Resolution-based decisison procedures for subclasses of first-order logic. Forthcoming PhD thesis, Univ. d. Saarlandes, Germany.

Hustadt, U. and Schmidt, R. A. (1997a), An empirical analysis of modal theorem provers. To appear in *J. Appl. Non-Classical Logics*.

Hustadt, U. and Schmidt, R. A. (1997b), On evaluating decision procedures for modal logics, *Research Report MPI-I-97-2-003*, Max-Planck-Institut für Informatik, Saarbrücken.

Hustadt, U. and Schmidt, R. A. (1997c), On evaluating decision procedures for modal logics, *in* M. Pollack (ed.), *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI'97)*, Vol. 1, Morgan Kaufmann, pp. 202–207.

Hustadt, U., Schmidt, R. A. and Weidenbach, C. (1998), Optimised functional translation and resolution. To appear in *Proc. TABLEAUX'98*, *Lecture Notes in Computer Science*, Springer.

Jaffar, J. (1990), Minimal and complete word unification, *J. ACM* **37**(1), 47–85.

Jónsson, B. and Tarski, A. (1951 & 1952), Boolean algebras with operators, Part I & II, *Amer. J. Math.* **73 & 74**, 891–939 & 127–162.

Jouannaud, J.-P. and Kirchner, C. (1991), Solving equations in abstract algebras: A rule-based survey of unification, *in* J.-L. Lassez and G. Plotkin (eds), *Computational Logic: Essays in Honor of Alan Robinson*, MIT-Press, pp. 257–321.

Joyner Jr., W. H. (1976), Resolution strategies as decision procedures, *J. ACM* **23**(3), 398–417.

Kirchner, C. and Klay, F. (1990), Syntactic theories and unification, *in* J. C. Mitchell (ed.), *Proc. LICS'90*, IEEE Computer Society Press, Philadelphia, pp. 270–277.

Klay, F. (1991), Undecidability properties of syntactic theories, *in* R. V. Book (ed.), *Rewriting Techniques and Applications: Proc. RTA'91*, Vol. 488 of *Lecture Notes in Computer Science*, Springer, pp. 136–149.

Ladner, R. E. (1977), The computational complexity of provability in systems of modal propositional logic, *SIAM J. Computing* **6**(3), 467–480.

Leitsch, A. (1997), *The Resolution Calculus*, EATCS Texts in Theoretical Computer Science, Springer.

Massacci, F. (1994), Strongly analytic tableaux for normal modal logics, *Proc. CADE-12*, number 814 in *Lecture Notes In Artificial Intelligence*, Springer, pp. 723–737.

McCune, W. (1994), OTTER 3.0 reference manual and guide, *Technical Report ANL-94/6*, Argonne National Lab., Argonne, IL.

Mendelson, E. (1987), *Introduction to Mathematical Logic*, third edn, Wadsworth & Brooks/Cole, Pacific Grove, California.

Mints, G. (1989), Resolution calculi for modal logics, *Amer. Math. Soc. Transl.* **143**, 1–14.

Mints, G. (1990), Gentzen-type systems and resolution rules. Part I: Propositional logic, *Proc. COLOG-88*, Vol. 417 of *Lecture Notes in Computer Science*, Springer, pp. 198–231.

Mohr, E. (1995), *Resolution-based calculi for modal logics*, Diplomarbeit, Univ. d. Saarlandes, Germany.

Morgan, C. G. (1976), Methods for automated theorem proving in nonclassical logics, *IEEE Trans. Comput.* **C-25**(8), 852–862.

Nivela, P. and Nieuwenhuis, R. (1993), Saturation of first-order (constrained) clauses with the *Saturate* system, *in* C. Kirchner (ed.), *Proc. RTA'93*, Vol. 690 of *Lecture Notes in Computer Science*, Springer, pp. 436–440.

Ohlbach, H. J. (1988a), *A Resolution Calculus for Modal Logics*, PhD thesis, Univ. Kaiserslautern, Germany.

Ohlbach, H. J. (1988b), A resolution calculus for modal logics, *in* E. Lusk and R. Overbeek (eds), *Proc. CADE'88*, Vol. 310 of *Lecture Notes in Computer Science*, Springer, pp. 500–516.

Ohlbach, H. J. (1991), Semantics based translation methods for modal logics, *J. Logic Computat.* **1**(5), 691–746.

Ohlbach, H. J. (1993a), Optimized translation of multi modal logic into predicate logic, *in* A. Voronkov (ed.), *Proc. LPAR'93*, Vol. 698 of *Lecture Notes in Artificial Intelligence*, Springer, pp. 253–264.

Ohlbach, H. J. (1993b), Translation methods for non-classical logics: An overview, *Bull. IGPL* **1**(1), 69–89.

Ohlbach, H. J., Gabbay, D. and Plaisted, D. (1994), Killer transformations, *Research Report MPI-I-94-226*, Max-Planck-Institut für Informatik, Saarbrücken.

Ohlbach, H. J. and Schmidt, R. A. (1997), Functional translation and second-order frame properties of modal logics, *J. Logic Computat.* **7**(5), 581–603. Also available as Research Report MPI-I-95-2-002, Max-Planck-Institut für Informatik, Saarbrücken.

Ohlbach, H. J., Schmidt, R. A. and Hustadt, U. (1995), Symbolic arithmetical reasoning with qualified number restrictions, *in* A. Borgida, M. Lenzerini, D. Nardi and B. Nebel (eds), *Proc. Intern. Workshop on Description Logics'95*, Vol. 07.95 of *Rap.*, Dipartimento di Informatica e Sistemistica, Univ. degli studia di Roma, Rome, pp. 89–95.

Ohlbach, H. J., Schmidt, R. A. and Hustadt, U. (1996), Translating graded modalities into predicate logic, *in* H. Wansing (ed.), *Proof Theory of Modal Logic*, Vol. 2 of *Applied Logic Series*, Kluwer, pp. 253–291. Also available as Research Report MPI-I-95-2-008, Max-Planck-Institut für Informatik, Saarbrücken (May 1995).

Otten, J. and Kreitz, C. (1996), T-string unification: Unifying prefixes in non-classical proof methods, *in* P. Miglioli, U. Moscato, D. Mundici and M. Ornaghi (eds), *Proc. TABLEAUX'96*, Vol. 1071 of *Lecture Notes in Artificial Intelligence*, Springer, pp. 244–260.

Plaisted, D. A. and Greenbaum, S. (1986), A structure-preserving clause form translation, *J. Symbolic Computat.* **2**, 293–304.

Plotkin, G. (1972), Building-in equational theories, *in* B. Meltzer and D. Michie (eds), *Machine Intelligence 7*, American Elsevier, New York, pp. 73–90.

Prior, A. N. (1968), Egocentric logic, *Nôus* **2**, 191–207.

Purdy, W. C. (1996a), Decidability of fluted logic with identity, *Notre Dame J. Formal Logic* **37**(1), 84–104.

Purdy, W. C. (1996b), Fluted formulas and the limits of decidability, *J. Symbolic Logic* **61**(2), 608–620.

Robinson, J. A. (1965), A machine-oriented logic based on the resolution principle, *J. ACM* **12**(1), 23–41.

Rock, G. (1995), *Transformations of first-order formulae for automated reasoning*, Diplomarbeit, Univ. d. Saarlandes, Germany.

Sahlqvist, H. (1975), Completeness and correspondence in the first and second order semantics for modal logics, *in* S. Kanger (ed.), *Proc. 3rd Scandinavian Logic Symposium, 1973*, North-Holland, pp. 110–143.

Schild, K. (1991), A correspondence theory for terminological logics: Preliminary report, *Proc. IJCAI'91*, pp. 466–471.

Schmidt, R. A. (1997a), Decidability by resolution for propositional modal logics. Submitted to *J. Automated Reasoning.*

Schmidt, R. A. (1997b), Relational grammars for knowledge representation. To appear in Böttner, M. (ed.), *Proc. of the Workshop on Variable-Free Semantics*, Fachbereich Sprach- und Literaturwissenschaft, Univ. Osnabrück.

Schmidt, R. A. (1997c), Resolution is a decision procedure for many propositional modal logics. To appear in Kracht, M., de Rijke, M., Wansing, H. and Zakharyaschev, M. (eds), *Advances in Modal Logic*, CSLI Publications, Stanford.

Schmidt, R. A. (1997d), Resolution is a decision procedure for many propositional modal logics, *Research Report MPI-I-97-2-002*, Max-Planck-Institut für Informatik, Saarbrücken.

Schmidt, R. A. (1998), E-unification for subsystems of S4. To appear in *Proc. RTA'98*, *Lecture Notes in Computer Science*, Springer.

Schmidt-Schauß, M. and Smolka, G. (1991), Attributive concept description with complements, *Artificial Intelligence* **48**, 1–26.

Schulz, K. U. (1992), Makanin's algorithm for word equations: Two improvements and a generalization, *in* K. U. Schulz (ed.), *Word Equations and Related Topics (Proc. IWWERT'90)*, Vol. 572 of *Lecture Notes in Computer Science*, Springer, pp. 85–150.

Simmons, H. (1994), The monotonous elimination of predicate variables, *J. Symbolic Computat.* **4**(1), 23–68.

Sperner, E. (1928), Ein Satz über Untermengen einer endlichen Menge, *Math. Z.* **27**, 544–548.

Stickel, M. E. (1985), Automated deduction by theory resolution, *J. Automated Reasoning* **1**, 333–356.

Szałas, A. (1993), On the correspondence between modal and classical logic: An automated approach, *J. Logic Computat.* **3**(6), 605–620. Also available as Research Report MPI-I-92-240, Max-Planck-Institut für Informatik, Saarbrücken.

Tseitin, G. S. (1970), On the complexity of derivations in propositional calculus, *in* A. O. Slisenko (ed.), *Studies in Constructive Mathematics and Mathematical Logic, Part II*, Consultants Bureau, New York, pp. 115–125. Reprint in Siekmann, J. and Wrightson, G. (eds) (1983), *Automation of Reasoning: Classical Papers on Computational Logic*, Vol. 2, Springer, 466–483.

van Benthem, J. (1983), *Modal Logic and Classical Logic*, Atlantic Heights: Bibliopolis, The Humanities Press, Napoli.

van Benthem, J. (1984), Correspondence theory, *in* D. Gabbay and F. Guenther (eds), *Handbook of Philosophical Logic*, Vol. II, Reidel, pp. 167–247.

van Benthem, J. (1993), Beyond accessibility: Functional models for modal logic, *in* M. de Rijke (ed.), *Diamonds and Defaults*, Kluwer.

van der Hoek, W. (1992), On the semantics of graded modalities, *J. Appl. Non-Classical Logics* **2**(1), 81–123.

van der Hoek, W. and de Rijke, M. (1993a), Counting objects in generalized quantifier theory, modal logic, and knowledge representation, *in* J. van der Does and J. van Eijck (eds), *Generalized Quantifiers: Theory and Applications*, ILLC, Univ. Amsterdam.

van der Hoek, W. and de Rijke, M. (1993b), Generalized quantifiers and modal logic, *J. Logic, Language and Inform.* **2**, 19–58.

van der Hoek, W. and de Rijke, M. (1995), Counting objects, *J. Logic Computat.* **5**(3), 325–345.

Viganò, L. (1997), *A Framework for Non-Classical Logics*, PhD thesis, Univ. d. Saarlandes, Germany.

Walther, C. (1987), *A Many-Sorted Calculus Based on Resolution and Paramodulation*, Research Notes in Artificial Intelligence, Pitman, London.

Weidenbach, C. (1997), Spass: Version 0.49, *J. Automated Reasoning* **18**, 247–252.

Weidenbach, C., Gaede, B. and Rock, G. (1996), SPASS & FLOTTER, version 0.42, *in* M. A. McRobbie and J. K. Slaney (eds), *Automated Deduction: CADE-13*, Vol. 1104 of *Lecture Notes in Artificial Intelligence*, Springer, pp. 141–145.

Zamov, N. K. (1989), Modal resolutions, *Soviet Math.* **33**(9), 22–29. Translated from *Izv. Vyssh. Uchebn. Zaved. Mat.* **9** (328) (1989) 22–29.

Appendix A

Eliminating second-order quantification with SCAN

Gabbay and Ohlbach (1992) developed the SCAN algorithm as a general tool for eliminating second-order quantification, that is particularly useful for doing modal correspondence theory automatically.[1] In this thesis we employ SCAN for computing the relational and functional correspondence properties for modal schemas. The analysis of the computation of SCAN led to the discovery of axioms, that are ordinarily not first-order definable in the relational setting, but have first-order functional correspondence properties by the optimised functional translation method (discussed in Chapter 3).

SCAN reduces existentially quantified second-order sentences to equivalent first-order formulations. Given a second-order sentence the algorithm generates sufficiently many logical consequences keeping from the resulting set of formulae only those in which no second-order variables occur. There are three stages that involve: (i) *Skolemisation*, (ii) *C-resolution* and (iii) *reverse Skolemisation*.

As input SCAN takes second-order formulae of the form

$$\psi = \exists P_1 \ldots \exists P_k \ \psi,$$

where P_i are existentially quantified predicate variables and ψ is a first-order formula. (SCAN can also be applied to universally quantified ψ, because negation will yield an existentially quantified formula that the algorithm can then manipulate. Negating SCAN's output produces the result for universally quantified formulae.) In the first stage, the Skolemisation stage, SCAN transforms a given existentially quantified second-order formula into clausal form via negation normal form and Skolemisation.

In the second stage, SCAN performs a special kind of constraint resolution, called *C-resolution*. It generates all and only resolvents and factors with the second-order variables that are to be eliminated. In the process the algorithm produces equations and inequations. The calculus is defined by the following two rules plus purity deletion.

$$\textbf{\textit{C}-resolution} \quad \frac{C \vee P(s_1, \ldots, s_n) \qquad D \vee \neg P(t_1, \ldots, t_n)}{C \vee D \vee s_1 \neq t_1 \vee \ldots \vee s_n \neq t_n}$$

$$\textbf{\textit{C}-factoring} \quad \frac{P(s_1, \ldots, s_n) \vee P(t_1, \ldots, t_n) \vee C}{P(s_1, \ldots, s_n) \vee C \vee s_1 \neq t_1 \vee \ldots \vee s_n \neq t_n.}$$

[1] SCAN is short for '*s*ynthesizing *c*orrespondence *a*xioms for *n*ormal logics'.

The conclusions are referred to as a *C-resolvent upon P* and a *C-factor upon P*, respectively. All possible C-resolvents and C-factors upon the predicate variables $P_1, \ldots, P_n$ are generated. When *all* C-resolvents and C-factors upon a particular literal and the rest of the clause set have been generated, purity deletion removes all clauses in which this literal occurs.

If the C-resolution stage terminates this yields a formula in which the specified second-order variables are eliminated. This formula is equivalent to the second-order formula originally put into the algorithm. If no clauses remain after purity deletion, then ψ is a tautology. And if C-resolution produces the empty clause, then ψ is unsatisfiable.

In the third stage, SCAN takes the subset of the generated clauses remaining after purity deletion (in which the variables P_i have been eliminated successfully), in the case that it is non-empty, and attempts to reconstruct the quantifiers for the Skolem functions by reversing Skolemisation. Reversing Skolemisation is not always possible, in particular, it is not, if the input formula is not first-order definable.

If the input formula is first-order definable then provided SCAN terminates the result is a first-order equivalent. If the input formula is not first-order definable and stage two terminates successfully bearing a non-empty set not containing the empty clause then SCAN produces equivalent second-order formulae in which the specified second-order variables are eliminated but quantifiers involving functions occur. These typically involve Henkin quantifiers.

SCAN is not complete for arbitrary second-order formulae. By a *complete* quantifier elimination algorithm we mean an algorithm that is guaranteed to find the equivalent first-order formula if there is one. In general, no complete algorithm exists, for otherwise arithmetic is enumerable. For the particular case of second-order formulae arising from the relational translation of modal axioms, it is an open problem whether a complete algorithm exists.

Appendix B

Orderings and ordered E-resolution

B.1 Orderings on terms, atoms, literals and clauses

A *(strict) ordering*, denoted by $\prec$, is an irreflexive and transitive binary relation. A strict ordering over a set X is *total* if for any pair of distinct elements $x, y \in X$, either $x \prec y$ or $y \prec x$. In general, a relation R over a set X is *well-founded* iff there is no infinite sequence $(x_i)_{i \in \mathbb{N}}$ of distinct elements of X such that for any $i \in \mathbb{N}$, $(x_{i+1}, x_i) \in R$.[1] In particular, a strict ordering $\prec$ over X is well-founded iff there is no infinite descending sequence such that $x_1 \succ x_2 \succ \ldots$. This is the case iff any subset of X has a $\prec$-minimal element.

We assume a given well-founded (mostly also total) strict ordering $\prec$ on the vocabulary. $\prec$ is called the *precedence*. We will give two constructions that lift the precedence to well-founded strict orderings among compound terms, atoms, literals or clauses with a common root.

The first construction employs the lexicographic ordering. It defines an ordering on tuples of fixed length n. Let $\{(X_i, \prec_i)\}_{i \in \{1, \ldots, n\}}$ be a finite family of strictly ordered sets. The *lexicographic ordering* over this family is the lexicographic combination of the orderings that is defined to be the product $(X_1 \times \ldots \times X_n, \prec)$ ordered as specified by

$$(x_1, \ldots, x_n) \prec (y_1, \ldots, y_n),$$

provided (i) $x_1 \prec_1 y_1$, or (ii) some k exists such that for any i below k, $x_i = y_i$ and $x_k \prec_k y_k$. A lexicographic ordering is well-founded (total) iff all its component-wise orderings are well-founded (total). More specifically, lifting the precedence to compound terms requires the lexicographic order $(X^n, \prec_{\text{lex}})$ of the nth power of a strictly ordered set $(X, \prec)$. Let f be an n-ary function on terms. f is said to have *lexicographic status* if $\prec^f$ is defined by:

$$(s_1, \ldots, s_n) \prec^f (t_1, \ldots, t_n)$$

if

(i) a permutation π exists on $\{1, \ldots, n\}$ with $(s_{\pi(1)}, \ldots, s_{\pi(n)}) \prec_{\text{lex}} (t_{\pi(1)}, \ldots, t_{\pi(n)})$ and

(ii) for any i, $s_i \prec f(t_1, \ldots, t_n)$.

[1] Dershowitz and Jouannaud (1990) say R is *terminating*.

Since constants are nullary functions they do not have status according to this definition.

The second construction employs the more general concept of a multi-set ordering. The multi-set ordering will primarily be used for defining an ordering on clauses from a strict ordering on literals. Remember we regard a *clause* as being a multi-set of literals. Let X be a set. A *multi-set* C over X is given by a function from X to the natural numbers. Intuitively, for any $x \in X$, $C(x)$ specifies the number of occurrences of x in C. We say, x is an element of C if $C(x)$ is non-zero. Let C and D be two multi-sets. Their union $C \cup D$ is defined by $(C \cup D)(x) = C(x) + D(x)$ and their intersection $C \cap D$ is defined by $(C \cap D)(x) = \min(C(x), D(x))$. We use the notation

$$\begin{aligned} C \vee D \quad &\text{for} \quad C \cup D, \\ C \vee L \quad &\text{for} \quad C \cup \{L\}, \end{aligned}$$

and for example

$$\begin{aligned} C = \neg A \vee B \vee B \quad \text{for} \quad & C(\neg A) = 1,\, C(B) = 2 \text{ and} \\ & C(L) = 0, \text{ for any other literal } L. \end{aligned}$$

A *multi-set ordering*, written $\prec_{\mathrm{mul}}$, on finite multi-sets over a set X is an extension of a strict ordering $\prec$ defined by: $C \prec_{\mathrm{mul}} D$ if $C \neq D$ and

$$\forall x \in C\ (D(x) < C(x) \rightarrow \exists y \in D\ (x \prec y \wedge C(y) < D(y))),$$

in words, for any x that occurs more often in C than in D, there is some y that is larger than x and occurs more often in D than in C. This definition ensures that, using the clause notation,

$$D \vee C_1 \vee \ldots \vee C_n \prec_{\mathrm{mul}} D \vee C,$$

if for any $i \in \{1, \ldots, n\}$ we have $C_i \prec C$. A multi-set ordering $\prec_{\mathrm{mul}}$ on finite multi-sets is well-founded (total) iff the underlying ordering $\prec$ is well-founded (total).

Consider again an n-ary function f on a set of terms strictly ordered by $\prec$. f is said to have *multi-set status* if $\prec^f$ is an ordering defined by: $(s_1, \ldots, s_n) \prec^f (t_1, \ldots, t_n)$ if

$$\{s_1, \ldots, s_n\} \prec_{\mathrm{mul}} \{t_1, \ldots, t_n\}.$$

The next step in lifting the precedence $\prec$ on the vocabulary to a well-founded strict ordering on the entire Herbrand universe[2] employs an instance of the concept of a recursive path ordering. Assume a precedence $\prec$ and assume each non-nullary function in the vocabulary has either lexicographic or multi-set status. A *recursive path ordering*, written $\prec_{\mathrm{rpo}}$, (over a precedence $\prec$ and the family $\{(\prec^f)\}_f$ of orderings) is defined by:

$$f(s_1, \ldots, s_m) \prec_{\mathrm{rpo}} g(t_1, \ldots, t_n)$$

provided one of three cases holds,

(i) for some i, $f(s_1, \ldots, s_m) \preceq_{\mathrm{rpo}} t_i$,

[2] the absolutely freely generated ground term algebra

(ii) if $f \prec g$ then $s_j \prec_{\text{rpo}} g(t_1, \ldots, t_n)$, for each j, or

(iii) if $f = g$ then $(s_1, \ldots, s_m) \prec^f_{\text{rpo}} (t_1, \ldots, t_m)$.

$\prec^f_{\text{rpo}}$ means $(\prec_{\text{rpo}})^f$. If each non-nullary function has lexicographic status and π is the identity function then the induced recursive path ordering is called a *lexicographic path ordering* and is denoted by $\prec_{\text{lpo}}$. If each non-nullary function has multi-set status then the induced recursive path ordering is called a *multi-set path ordering* and is denoted by $\prec_{\text{mpo}}$.

The final step is lifting the ordering of the Herbrand universe to the set of literals and the set of clauses over the Herbrand universe. The ordering on the atom set is defined from the ordering on the Herbrand universe by a recursive path ordering (either a lexicographic path ordering of a multi-set path ordering). We let the ordering on literals be given by the multi-set ordering on a multi-set encoding of literals. Any positive literal L is encoded by a singleton set { L } and any negative literal by the set $\{L^\neg, L^\neg\}$ with two copies of the complementary literal of L. The ordering on the set of clauses is then determined by the multi-set ordering of the ordering on the literal encodings.

When the precedence on the function symbols and the predicate symbols is a well-founded total strict ordering then so is the ordering on the set of ground clauses.

Orderings on ground expressions (terms, atoms, literals and clauses) lifts to non-ground expressions by the definition: $s \prec t$ if for any substitution σ, $s\sigma \prec t\sigma$.

B.2 Completeness of ground ordered E-resolution

In this section we prove Theorem 7.1.1.

To begin with we supply some preliminary definitions. A *Herbrand model* (or *interpretation*) I is a set of ground atoms and the presence of an atom A in I means A is true in I and $\neg A$ is false in I. In general, a clause $L_1 \vee \ldots \vee L_n$ is true in I iff some positive $L_i = A_i$ is in I or for some negative literal $L_i = \neg A_i$, A_i is not in I. Falsehood is defined dually. As we are concerned with E-resolution and we will construct an E-model, we relativise the definition of truth: An atom A is *E-satisfiable* in I iff I contains some atom E-equivalent to A (and similarly for clauses).

The hard direction of the completeness theorem is proving that for any unsatisfiable set S of ground clauses there is a deduction of the empty clause from S. Bachmair and Ganzinger (1994, 1997) prove this by constructing a certain Herbrand model I_S for any S that is closed under ground deduction and does not contain the empty clause. The construction of I_S is such that, any clause C is in S iff it is E-satisfiable in I_S, and this implies any E-unsatisfiable saturated set S contains a contradiction.

The idea of the construction of the Herbrand model I_S is the following. Suppose S is a set of ground clauses. Define I_S inductively with respect to $\prec$, by starting with the minimal clauses of S, identifying maximal ground atoms and iteratively adding the associated smallest equivalent atoms to partial models I_C forming minimal extensions. This process is iterated for $S \setminus \{C\}$, and so forth. Because $\prec$ is admissible and E-compatible, every E-congruence class $A/{=_E}$ of ground atoms has a smallest element, which we denote by $\inf_\prec(A/{=_E})$.[3] These will function as representatives in the I_C. Formally, define ε_C and

[3]When E forms a convergent rewrite system oriented by $\prec$, then $\inf_\prec(A/{=_E})$ coincides with $N_E(A)$.

I_C by

$$\varepsilon_C = \begin{cases} \{\inf_{\prec}(A/{=_E})\} & \text{provided (i) } A \text{ is maximal in } C\text{, (ii) } \inf_{\prec}(A/{=_E}) \notin I_C\text{, (iii)} \\ & C \text{ is } E\text{-unsatisfiable in } I_C\text{, and} \\ \emptyset & \text{otherwise} \end{cases}$$

and

$$I_C = \bigcup\{\varepsilon_D \mid D \prec C\}.$$

I_C is intended to be an E-model of the clauses (in S) strictly smaller than C, and ε_C is a minimal extension such that C is E-satisfiable in $I_C \cup \varepsilon_C$. The intended Herbrand model I_S of S is defined by

$$I_S = \bigcup\{\varepsilon_C \mid C \in S\}.$$

A clause C for which ε_C is non-empty is called a *productive* clause. We say C *produces* $\inf_{\prec}(A/{=_E})$.

Recall, ground deduction is defined by:

Ground ordered E-resolution $\dfrac{C \vee A_1 \vee \ldots \vee A_n \qquad C' \vee \neg A}{C \vee C'}$

provided (i) $A_1 =_E \ldots =_E A_n =_E A$, (ii) no atom in C is E-equivalent to any A_i, (iii) every atom in C is strictly smaller than any A_i with respect to $\prec$, and (iv) $\neg A$ is $\prec$-maximal in $C' \vee \neg A$.

This inference rule is sound, because the conclusion is a logical consequence of its premises. The next lemma and theorems prove completeness for ground inference. The proofs are very similar to the proofs for ordered resolution under the empty theory, and parallel the proofs found in Bachmair and Ganzinger (1997) for general resolution (Lemma 3.3, Theorem 3.4 and Lemma 5.2). To avoid cluttering we will write 'true' when we mean 'E-satisfiable' and 'false' when we mean 'E-unsatisfiable'.

Lemma B.2.1 Let S be a set of ground clauses. Any productive clause C in S is E-satisfiable in I_S.

Proof. Since ε_C is non-empty and is contained in I_S, C is E-satisfiable in I_S. □

Theorem B.2.2 Let S be a saturated set of ground clauses (with respect to the above ground resolution rule) and the empty clause is not in S. Then I_S is an E-model of S.

Proof. The proof is by contradiction. Because every productive clause in S is true in I_S (by the lemma), we assume there are non-productive clauses in S that are not true in I_S. Then there is a smallest such clause. Let C be the smallest non-productive clause in S that is not true in I_S. Suppose A is the maximal atom in C. We distinguish two cases.

(i) $\inf_{\prec}(A/{=_E}) \in I_S$: Let $C = C' \vee \neg A$. Suppose D in S produces $\inf_{\prec}(A/{=_E})$. Then $D = D' \vee A_1 \vee \ldots \vee A_n$, for $\{A_1, \ldots, A_n\}$ the largest set of atoms occurring in D' that is included in $A/{=_E}$. This set is maximal in $D/{=_E}$ with respect to $\prec^E$ and every atom in D'

is strictly smaller than any A_i with respect to $\prec$. C and D resolve and produce $C' \vee D'$ which is in S. Evidently, $C' \vee D' \prec C$. We get a contradiction if we show that $C' \vee D'$ is false in S. Now, as C is false in I_S, C' is also false in I_S. It remains to prove D' is false in I_S. By the definition of a productive clause, D is false in I_D. Hence, D' is false in I_D. This implies D' is false in I_S. For, suppose not. Two situations are possible.

a. Assume $D' = D'' \vee B$ with B maximal in D'. B is smaller than A. Then, since D' is true in I_S, $\inf_\prec(B/{=_E})$ is in I_S. Hence there is a clause C that produces $\inf_\prec(B/{=_E})$ and $C \prec D$. However, then $\inf_\prec(B/{=_E}) \in I_D$ and D is true in I_D which contradicts one of the conditions of D being productive.

b. Assume $D' = D'' \vee \neg B$ with $\neg B$ maximal in D'. D' is true in I_S implies $\inf_\prec(B/{=_E}) \notin I_S$, which in turn implies $\inf_\prec(B/{=_E}) \notin I_D$, hence D is true in I_D. Again, this is a contradiction.

(ii) $\inf_\prec(A/{=_E}) \notin I_S$: No A' E-equivalent to A occurs negatively in C, for otherwise C is true in I_S. Suppose C' is such that $C = C' \vee A_1 \vee \ldots \vee A_n$ and the set $\{A_1, \ldots, A_n\}$ the largest set of atoms occurring in C that is included in $A/{=_E}$. A is maximal in the set $\{A_1, \ldots, A_n\}$. $\inf_\prec(A/{=_E})$ is not in I_C, else $\inf_\prec(A/{=_E}) \in I_S$ which contradicts our assumption. Now, we prove C is false in I_C, since then C is true in $I_C \cup \{\inf_\prec(A/{=_E})\}$, and therefore, C is productive which contradicts that $\inf_\prec(A/{=_E}) \notin I_S$. We assume C is true in I_C and consider two cases.

a. B occurs positively in C' and $\inf_\prec(B/{=_E}) \in I_C$. Hence $\inf_\prec(B/{=_E}) \in I_S$, and both C' and C are true in I_S, which is a contradiction.

b. No atom B with $\inf_\prec(B/{=_E}) \in I_C$ occurs positively in C'. Suppose $C' = C'' \vee \neg B$ and $\neg B$ is maximal in C'. $\inf_\prec(B/{=_E}) \notin I_C$. Since C is false in I_S there is a clause D in S that produces $\inf_\prec(B/{=_E})$. By assumption, $B \prec A$. This implies $D \prec C$, and consequently, $\inf_\prec(B/{=_E}) \in I_C$, which is a contradiction. □

Completeness of ground inference without redundancy follows. With redundancy the proof is more technical.

Theorem B.2.3 Let S be a set of ground clauses and $\prec$ an admissible ordering on atoms compatible with E. Let S be saturated up to redundancy with respect to ground ordered E-resolution. Then, for any atom A and any clause C in S with maximal atom A,

(i) if C produces $\inf_\prec(A/{=_E})$ then C is not redundant, and for any clause D in S that is strictly smaller than C, D is true in I_C, and

(ii) C is true in I_S.

Proof. Case (i): The proof is by induction determined by the ordering $\prec$. Let A be any atom. Assume for any $B \prec A$ and any D with maximal atom B the properties (i) and (ii) hold.

Suppose C is a clause with maximal atom A and produces $\inf_\prec(A/{=_E})$. First, we prove C is not redundant. If C is redundant then either $\models C$ or there are n clauses $C_1, \ldots, C_n$ in S $(n > 0)$ such that $C_i \prec C$ for each i and $\models (C_1 \wedge \ldots \wedge C_n) \to C$. In the case that $\models C$, C is a tautology and is true in all models, in particular, also in I_C. This contradicts one of the conditions that C is productive. In the alternative case, since C is false in I_C, one of the C_i is also false in I_C. However, according to the definition of I_C, C_i is true in I_C.

Second, we let D be any clause in S with $D \prec C$ and show D is true in I_C by considering three situations.

a. The maximal atom of D is strictly smaller than A. By (ii) of the inductive hypothesis D is true in I_S. This is the case, when 1. B is an atom that occurs positively in D and $\inf_{\prec}(B/{=_E}) \in I_S$, or when 2. no atom of D has its smallest E-equivalent in I_S. In the first case, $\inf_{\prec}(B/{=_E})$ is produced by either D or some clause $D' \in S$ smaller than D. In either situation $\inf_{\prec}(B/{=_E}) \in I_C$, which implies D is true in I_C. In the second case, every atom of D occurs negatively therein and no atom of D has its smallest E-equivalent in I_C. Hence D is true in I_C.

b. A occurs in D and D is redundant in S. This means there are clauses $D_1, \ldots, D_n \in S$ $(n \geq 0)$ with $D_i \prec D$ for each i and $\models (D_1 \wedge \ldots \wedge D_n) \rightarrow D$. By the inductive hypothesis each D_i is true in I_S. By an argument like the one in a. for D, each D_i is also true in I_C. Therefore, D is true in I_C (using modus ponens).

c. A occurs in D and D is not redundant in S. We assume D is false in I_C and derive a contradiction. A does not occur positively in D. Suppose $D = D' \vee \neg A$. Suppose $C = C' \vee A_1 \vee \ldots \vee A_n$ with the restriction that no atom in C' is E-equivalent to any A_i. $C' \vee D'$ is a resolvent of C and D. Inferences with non-redundant clauses in saturated sets are redundant, and this means, $C' \vee D'$ is redundant with respect to S. Then there are clauses $D_1, \ldots, D_n \in S$ $(n \geq 0)$ with $D_i \prec C' \vee D'$ for every i and $\models (D_1 \wedge \ldots \wedge D_n) \rightarrow D$. D' is false in I_C (since D is false in I_C and A does not occur positively in D'). C' is false in I_C (since C produces A and A does not occur positively in C'). Consequently, one of the D_i must also be false in I_C. But, since $D_i \prec C' \vee D' \prec D$, D_i is true in I_C, by the definition of I_C.

Case (ii): Our goal is to show C is true in I_S, which we do by induction. The inductive hypothesis is that all $D \in S$ strictly smaller than C are true in I_S. If C is redundant in S then clearly C is true in I_S, because the clauses smaller than C that imply C are all true in I_S. Also, if C is productive then it follows easily that C is true in I_S. The difficult part is proving the result for when C is both non-redundant and non-productive. Suppose C is false in I_S. Let A be the maximal atom of C.

a. $\inf_{\prec}(A/{=_E}) \in I_S$: Then A does not occur positively in C. Assume $C = C' \vee \neg A$. Suppose D produces $\inf_{\prec}(A/{=_E})$ and $D = D' \vee A_1 \vee \ldots \vee A_n$ such that $C' \vee D'$ is the resolvent of C and D. If $C' \vee D'$ belongs to S then we obtain a contradiction as in Theorem B.2.2 (i). The inference of $C' \vee D'$ is redundant, and hence $C' \vee D'$ is redundant in S. Then, as before, there are clauses $D_1, \ldots, D_n \in S$ $(n \geq 0)$ with $D_i \prec C' \vee D'$ for every i and $\models (D_1 \wedge \ldots \wedge D_n) \rightarrow D$. C is false in I_S then so is C'. D' is false in I_S. Therefore, some D_i is false in I_S. But, by the inductive hypothesis D_i is true in I_S.

b. $\inf_{\prec}(A/{=_E}) \notin I_S$: This means no atom A' E-equivalent to A occurs negatively in C and it follows $C = C' \vee A$. C satisfies all the properties of being productive, which contradicts our assumption. $\square$

The essential property used in both theorems is that, if the subclause D' of a productive clause $D = D' \vee A$ is false in I_D then D' is false in I_S. For the property to hold, it is crucial that $\prec$ is compatible with E, and the congruence classes are ordered by $\prec^E$.

List of figures

Index of notation

Index of schemas, rules and logics

Subject index