Analytic Proof Systems for Classical and Modal Logics of Restricted Quantification

Ian P. Gent[1]

Department of Computer Science
University of Warwick
Coventry CV4 7AL
United Kingdom

March 1993

[1]Current address:
Department of Artificial Intelligence
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN
United Kingdom
Email: I.P.Gent@ed.ac.uk

Author's Note

This paper is my PhD thesis as submitted to the University of Warwick in March 1992. It was formatted using the Unix programs `tbl`, `eqn`, and `psroff` (a version of `troff`). Since then, the department of computer science has changed the version of `troff` it uses, and I have changed to using LaTeX.[1] The result is that the only corrections I have made here are those that can be achieved easily by editing the formatted files, and I have included a small number of other errata.

I would like to take this opportunity of thanking Hubert Gent, Margaret Gent, David Gent, Ann Gent, Lydia Brimage, Kathy Courtney, Félix Hovsepian, David Randell, Guy Saward, Mike Slade, Chris Ramsden, Nick Holloway, Tony Cohn, Meurig Beynon, Alan Bundy, Lincoln Wallen, Rajeev Goré, Marcello D'Agostino, Harrie de Swart, Alan Frisch, Richard Scherl, Brian McConnell, Uwe Petermann, Wilfred Chen, John Derrick, the Science and Engineering Research Council, the Computer Science Department of the University of Warwick, the Society for Exact Philosophy, NATO, the Mathematical Sciences Institute at Cornell University, Harold Simmons, John Tucker, Iain Alexander-Craig, Henry Kautz, and Judith Underwood for many different types of invaluable help. As has been said:

> *A project of this scope could not be realized without the aid of many people, or rather it* could *but it would be dumb to do it that way when there are so many people around willing to give their aid.*
>
> — Peter Schickele
> *The Definitive Biography of P.D.Q. Bach (1807-1742)?*

Ian Gent
Department of Artificial Intelligence, University of Edinburgh
March 1993

[1] which does *not* mean that I *like* LaTeX.

Errata

- page 61, line 1
  "first" should be "second"

- page 61, line 5
  "second" should be "first"

- page 63, paras (1)(i) and (1)(ii)
  The text of these two paragraphs should be reversed.

- page 78, Theorem 8.2.2
  "Σ**InstTab**(*1345*)" is omitted from the statement of the theorem.

- page 78, sentence on last two lines
  "Σ**Tab**" should be "Σ**InstTab**" throughout this sentence (3 occurrences).

Analytic Proof Systems for Classical and Modal Logics of Restricted Quantification

Ian Gent

Abstract

This thesis is a study of the relationship between proof systems for propositional logic and for logics of restricted quantification incorporating restriction theories. Such logics are suitable for the study of special purpose reasoning as part of a larger system, an important research topic in automated reasoning. Also, modal and sorted logics can be expressed in this way. Thus, results on restricted quantification apply to a wide range of useful logics. D'Agostino's "expansion systems" are used to generalise results to apply to a variety of tableau-like propositional proof systems.

A certain class of propositional expansion systems is defined, and extended for restricted quantification in two different ways. The less general, but more useful, extension is proved sound and complete provided that the restriction theory can be expressed as a set of definite Horn clauses.

In the definite clauses case, the result is used to present a generalisation of Wallen's matrix characterisations of validity for modal logics. The use of restricted quantification enables more logics to be covered than Wallen did, and the use of expansion systems allows analogues of matrices to be defined for proof systems other than tableaux.

To derive the results on matrices, the calculi for restricted quantification are made weaker, and so can be unsound for some restriction theories. However, much greater order independence of rule applications is obtained, and the weakening is sound if one of two new conditions introduced here hold, namely "alphabetical monotonicity" or "non-vacuity". Alphabetical monotonicity or non-vacuity are shown to hold for a range of interesting restriction theories associated with order sorted logics and some modal logics.

I also show that if non-vacuity holds, then instantiation in restricted quantification can be completely separated from propositional reasoning.

The major problem left open by this thesis is whether analogues of the previous matrix characterisations can be produced based on the proof systems introduced for non-definite clause restriction theories.

TABLE OF CONTENTS

Chapter 1

Introduction

If your thesis is utterly vacuous
Use first order predicate calculus.
With sufficient formality
The sheerest banality
Will be hailed by the critics: "Miraculous!"

-- *Henry A Kautz*

An important way of improving the efficiency of automated reasoning is to pick out subproblems for which efficient inference methods are available, and to build such methods into more general reasoning processes. This thesis represents a foundational study of the interface between efficient, special purpose, inference and general inference. This is done by the study of the relation between proof systems for first order quantified logics and for propositional logic, the quantified logics I consider being logics using *restricted quantification*. This provides a framework in which formulas from a special purpose theory can be separated from the rest of a logic: in this thesis I show that this separation can be incorporated into proof systems, clearly isolating the special purpose reasoning.

The two main ideas of restricted quantification can be illustrated very quickly. The first idea is that the domain of each quantifier is restricted by a first order predicate. I will take a restricted quantifier to be either "$\forall x_{:\rho(x)}$", read as "for all x such that $\rho(x)$", or "$\exists x_{:\rho(x)}$", read as "there is some x such that $\rho(x)$". The new quantifiers do not add any expressiveness to the ordinary predicate calculus, because "$\forall x_{:\rho(x)}\ \phi(x)$" is understood to mean *exactly* the same as if it had been written "$\forall x\ (\rho(x) \supset \phi(x))$" while "$\exists x_{:\rho(x)}\ \phi(x)$" is understood to mean *exactly* the same as if it had been written "$\exists x\ (\rho(x) \wedge \phi(x))$". The second idea introduces special purpose theories: this is achieved by adding *restriction theories* to logics of restricted quantification. A restriction theory is simply a set of predicate calculus statements in the language of the predicates appearing as restrictions. Using restriction theories, both modal logics and sorted logics can be expressed very easily as logics of restricted quantification. For instance, in modal logics, we can use the simple fact that $\Box\phi$ is true in world ε iff $\forall w_{:\varepsilon \mathcal{R} x}\ \phi(x)$. Here, "$\mathcal{R}$" is the usual accessibility relation, and we vary its properties by varying the restriction theory. For example, the theory corresponding to **S4** expresses the transitivity and reflexivity of the restriction theory. Again this does not extend the expressiveness of restricted quantification beyond that of the predicate calculus, because I will take restriction theories to be first order.

Restricted quantification provides an excellent framework for the study of the link between special purpose inference methods and general methods. It gives us a a simple framework for expressing that one part of a logic is to be considered as a special purpose theory. As a result, it is possible to construct proof systems which very clearly isolate special purpose reasoning from general reasoning: doing this forms the subject of this thesis.

As mentioned above, modal logics can be expressed using restricted quantification. Automated reasoning for modal logics has been a considerable research subject in recent years. Despite the presence of the accessibility theory in the semantics of modal logics, most workers have used indirect methods to reason with this while those who have reasoned explicitly with the accessibility theory, such as Wallen (1989)

have done so only in the context of modal logics.[1] In this thesis I show how my results may be applied to modal logics: these results obtain all the advantages of earlier specially designed methods such as Wallen's while being much more general.

This thesis concentrates exclusively on the link between proof systems for propositional logic and proof systems for logics of restricted quantification. However, I do not seek to choose one propositional proof system, such as analytic tableaux or resolution, and work on first order extensions of that. I avoid the choice of a particular propositional proof system by using D'Agostino's (1990) definition of an *expansion system*, which generalises many proof systems. There is very little theoretical basis for the choice of a particular propositional proof system as optimal, given the lack of formal knowledge concerning the properties of different propositional systems for proof search: therefore it is better to discuss techniques that can be applied to a variety of propositional proof systems. I discuss this important point further in Section 1.2 of this introduction, since it is a point most workers in this field have missed, with the exception of Frisch (1991).

The first order proof systems presented in this thesis are all *ground* proof systems; that is, every application of every proof rule for quantification must introduce explicit terms. Proof systems introducing free variables in which unification is used are not considered. This is not to underestimate the importance of unification in automated reasoning. Instead, it is to claim that in my study of extending propositional proof systems for restricted quantification, unification is not an issue. The study of unification problems is a research topic in itself, and I could not do it justice in the scope of this thesis. My contributions can be much more effectively expressed than if I included particular unification algorithms, possibly using non optimal ones and confusing the contributions I make.

The results obtained in this thesis represent the isolation and generalisation of earlier insights which have been used to develop automated reasoning systems. This represents a notable contribution to the development of automated reasoning as a science. By generalisation of an insight, useful techniques can be applied to a wider range of problems. By isolation of an insight, the basis of a successful technique can be seen much more clearly and with less technical detail intervening between an efficient system and the understanding of that system. As the processes of isolation and generalisation develop in more subfields of automated reasoning, those research workers building reasoning systems can call on ever greater bodies of expertise. System designers can make more rational design decisions while having to do less theoretical work to justify those decisions.

In the rest of this introduction I discuss further the results, aims and methods of my work, and then present a chapter-by-chapter outline of this thesis.

Section 1.1: Results

The only new result in this thesis which concerns propositional logic is a rather minor and easy one. Namely, I show how to extend Mondadori's (1989) proof system **KE** to deal with propositional equivalences.

Apart from the extension of **KE**, all the other results in this thesis concern extending propositional expansion systems to deal with restricted quantification.

The main results of this thesis are as follows:

(1) I show how a certain class of propositional expansion system can be extended to a sound and complete proof system for restricted quantification, provided that the restriction theory has what I call the "least Herbrand model property". This simply demands that all sets of restrictions have a least Herbrand model[2] in the restriction theory. This condition is satisfied if the restriction theory

[1] Frisch and Scherl (see references) are most honourable exceptions to this. However they do not consider tableau-style or matrix-style proof systems, which are the main subject of this thesis.

[2] Least Herbrand models are also called "unique minimal models" and "initial models" elsewhere in the literature.

can be written in definite Horn clauses.[3] Such calculi have two interesting and crucial properties. First, no branching is introduced into proof trees by the special purpose reasoning associated with restricted quantification. Second, reasoning within the restriction theory takes place only when checking the side condition on the expansion rule for universal instantiation. Because of these properties these calculi form the foundation for the next four results, (2) to (5). The last result, (6), concerns the consequences of the least Herbrand model property failing.

(2) I show how the order dependence of rule applications due to the side condition on universal instantiation can be eliminated by the use of a weaker side condition. The resulting proof system is sound if one of two new conditions I identify hold, namely "alphabetical monotonicity" or "non-vacuity".

(3) I show that alphabetical monotonicity or non-vacuity holds for a range of interesting restriction theories, including those associated with order sorted logics and a selection of modal logics.

(4) I generalise Wallen's (1989) matrix characterisations of validity along two axes: I give the result if alphabetical monotonicity or non-vacuity holds; and I define analogues of matrices based on an arbitrary countermodel expansion system.

(5) I show that if non-vacuity holds for a restriction theory, then instantiation in restricted quantification can be completely separated from propositional reasoning.

(6) I show how to extend propositional countermodel systems to sound and complete systems when the least Herbrand model property does not hold. Unlike the proof systems considered at (1), the resulting proof systems allow branching when universal instantiation takes place. Whether or not analogues of the previous four results can be produced in this case represents the major problem left open by this thesis.

Section 1.2: Aims

Two main aims lie behind my work. The first is to examine proof systems for restricted quantification. The second is to do so by extending propositional proof systems, in as much generality as possible with respect to the propositional systems. The worth of both of these aims needs to be justified.

Restricted quantification, as presented in this thesis, allows one to separate out *syntactically* one kind of logical information (that from the restriction theory) from another (that of ordinary logical information). This has long been recognised as being useful ''syntactic sugar'' although it does not extend the expressiveness of a logic (see for instance Hailperin 1957a, 1957b). If a logic can be expressed by restricted quantification in such a way that the restriction theory is very simple, and all references to the restriction theory appear only in restrictions, then reasoning about restrictions can be efficiently implemented without full predicate logic reasoning. A particular use of this kind of separation has been made in sorted logics, where this type of syntactic separation can improve the efficiency of automated theorem proving (see for example Cohn's (1989a) survey paper). In sorted logics the theory is often extremely simple. This emphasises the difference between reasoning in the theory (perhaps taking linear or quadratic time) and reasoning outside the theory (which is undecidable).

Many logics of particular interest can be expressed very simply as logics of restricted quantification. Examples are modal logics and sorted logics. Modal logics have been studied because of the usefulness of modalities in expressing concepts such as knowledge and belief (see for example Halpern and Moses 1985). Many-sorted logics have been studied because their use can improve the efficiency of deduction (see for example Cohn 1989a). For these logics, many special purpose proof systems have been developed to take account of the separation of information between the accessibility theory and the rest of the logic (in modal logics) or between the sort theory and the rest of the logic (in sorted logics). Until the recent work of Frisch (1991) and Bürckert (1990a), no work in automated reasoning focused on the more

[3] Definite Horn clauses are clauses in which at exactly one positive literal appears.

general framework of restricted quantification. Therefore it has been difficult to appreciate the fundamental insights lying behind many of the proof systems developed for these logics. The work of Frisch and Bürckert has begun to show that the same insights used to develop proof systems for modal or sorted logics can be used in the general context. This thesis justifies this further. Therefore there is a simple reason for studying proof systems for restricted quantification: *we obtain a great deal of generality and lose very little efficiency in the resulting proof systems.*

In this thesis I do not extend particular propositional proof systems to restricted quantification: instead I show how whole classes of propositional proof systems may be extended. There are two reasons for this. First, many results about extending propositional proof systems to quantificational proof systems can be generalised so there is no point restricting the results. Second, the basic properties of propositional proof systems for proof search are very little known. Therefore any commitment to quantified systems based on a particular propositional system risks being the wrong commitment. The first point is simple, but I will discuss the second further in the rest of this section. It seems to have been missed by most authors, judging by the fact that most authors who deal with sorted logics, modal logics, or restricted quantification only consider extensions of particular propositional proof systems.

There is, sadly, very little knowledge concerning the relationship between proof search in different propositional proof systems. A body of fine work, starting with Cook and Reckhow (1974, 1979), has established some worst case results for individual proof systems and some results about the relationships between size of proofs in different proof systems. However, many of these results are not sufficient to relate the properties of proof *search* in different proof systems.

A typical example of a result relating proof size in different proof systems is given by Boolos (1984). He shows there is a class of formulas whose shortest proofs in a natural deduction system (that of Mates 1972) can be exponentially shorter than their shortest proofs in an analytic tableaux system. Unfortunately, even this kind of devastating result is not conclusive for proof search. Although we can be sure that the tableau proof cannot be found in better than exponential time, we cannot be sure that the natural deduction proof can be found in a shorter time by a general purpose theorem proving method.

As well as not being necessarily suitable for considerations of proof search, such results can be rather surprising. This means that judgements of the desirability of different proof systems must be based on only the most rigorous results, and such results are usually not available. A recent, startling, example relating truth tables and analytic tableaux, is due to D'Agostino (personal communication 1991, conjectured in D'Agostino 1990). It is easy to construct examples showing that the size of analytic tableaux can be exponentially smaller than truth tables, and it has been widely assumed that tableaux are a uniform improvement on truth tables. D'Agostino quotes noted authors who have implied this, including Beth (1958) (one of the inventors of tableaux) as well as introductory textbooks by Gallier (1987) and Jeffrey (1981). D'Agostino has shown that this assumption is mistaken: on certain classes of examples, analytic tableaux must be exponentially larger than the equivalent truth tables.

Unfortunately, formal results relating proof search in different proof systems are rarer than results on the maximum size of proofs. An example is again due to D'Agostino (1990). He shows that any proof search strategy for analytic tableaux can be simulated in Mondadori's (1989) system **KE**. Furthermore, the simulation can always be done in linear time, but in some cases leads to proofs very much more quickly than in the tableaux. Goré and D'Agostino (1991) show some empirical consequences of this result. Bibel (1982) also studies the theoretical relationship between various propositional theorem proving methods.

It is rather disturbing that theoretical uncertainties about the relationship between different proof systems appear for propositional logic. The situation for quantified logic can only be worse.

Faced with the lack of theoretical comparisons, it is tempting to think that we can turn to empirical tests of different automated theorem provers to form a judgement about their worth. Unfortunately we have no real insight how to construct valid empirical tests. Concentration on single examples such as Schubert's

steamroller (Stickel 1986) is rather pointless, while even such useful lists as Pelletier's (1986) list of seventy five problems beg the question of how they were constructed. Such examples, and other researchers' results on them, should certainly be used as guides if one is implementing an automated theorem prover, but they provide little basis for a strong belief in the superiority of one method over another. In this I strongly disagree with Wos's statement that

> ''Different reasoning programs and different approaches within a reasoning program should be compared and analyzed in detail by gathering statistics from attempts to answer questions and solve problems; the statistics concerning CPU seconds, of course adjusted for the type of computer being used, are especially significant.'' (Wos 1988, page 28, Research Principle 2)

I prefer instead, at least for judging the effectiveness of different theorem proving methods, Wallen's maxim that

> If you can prove that it's better, it's not worth implementing; and if you can't prove that it's better, it's not worth implementing. (Wallen, personal communication[4] 1989)

Unfortunately, whether or not one seeks empirical tests of theorem provers, no body of examples such as Pelletier's is available for restricted quantification. This can be clearly illustrated by considering modal logics. Standard examples of theorems from textbooks such as Hughes and Cresswell (1968) can be proved by rather simple methods such as Morgan's (1976) yet few more complex examples have arisen in the literature. Fitting and Mints have been pleading at conferences for such a set of examples,[5] but none appears to be forthcoming. Indeed, there is very little published material concerning implementations of modal logic theorem provers, even for a non standard set of examples. Even authors such as Fitting (1988) and Wallen and Wilson (1987), in papers explicitly discussing implementations, do not give examples of theorems that had been proved using their systems. There is at least some hope. Modal and temporal logics seem to be coming in to some favour as tools in artificial intelligence and theoretical computer science (see for example Atkinson and Cunningham 1991, Torsun and Manning 1990, Gent 1991a). These fields may well provide a range of non-trivial examples.

From this discussion it can be seen that there is no good reason to choose one propositional proof system over another, either in general or for extension to restricted quantification. Yet the choice of what might turn out to be the wrong system runs the risk of wasting a lot of work. So there are considerable advantages to be gained from avoiding premature choice of a propositional proof system. In this thesis I avoid this choice by using a general formalisation of refutation based proof systems, namely expansion systems.

Section 1.3: Methods

My aim has been to explore the link between proof systems for propositional logic and restricted quantification. I have tried to do this using methods of as much generality as possible.

First, D'Agostino's (1990) formalisation of expansion systems allows results to be generalised across a variety of propositional proof systems. Furthermore, it provides a sound basis for such general results, because one can easily express properties such as soundness and completeness of expansion systems. In this thesis I provide formal proofs of results for restricted quantification based on such properties of propositional expansion systems.

Starting with propositional expansion systems as my base, I show how these can be extended in a very simple way to proof systems for restricted quantification. This is the principal use in this thesis of a semantic argument for deriving proof systems for restricted quantification. By choosing a simple extension of propositional proof systems, the semantic argument is kept as simple as possible, and the resulting characterisation of validity is also as simple as possible. This then provides an excellent

[4] by telephone, hence the lack of quotation marks.

[5] At two separate conferences I have attended Fitting has made this plea, crediting it to Mints.

starting point for developing other proof systems for restricted quantification. Having used the semantic argument to give a relatively simple characterisation of validity, I subject the resulting proof systems to a considerable amount of syntactic, proof theoretical, analysis.

Syntactic arguments relate two different proofs, which are always finite objects. This can be a considerable advantage over semantic arguments, which must relate syntax to infinite objects such as models. This enables me to discuss a wide range of proof systems based on the ones derived semantically, with the intention of deriving more computationally sensitive characterisations of validity. Using this approach gives more insight than the bare results: the various conditions on the theorems in this thesis can all be seen to arise at particular places. Different conditions arise from what is required of propositional systems, of the restriction theory to extend these systems for restricted quantification, and again of the restriction theory in the derivation of later proof systems. These conditions arise separately, so the reasons for each can be seen separately and in perfect context.

In summary, my approach is to use just one semantic argument, deriving a possibly inefficient proof system, and then apply syntactic arguments to derive more efficient proof systems. This seems to me to be a very natural approach. The concept of computationally efficient proof systems is not directly related to the semantics of logics. It is, however, a relationship between different syntactic entities, namely proof systems. By the explicit use of syntactic arguments which relate different proof systems, this relationship is made clear. Indeed, these syntactic arguments are motivated by a desire to derive computationally efficient proof systems. In particular, the main motivation is to derive proof systems in which the order dependence between different parts of a proof is minimised. This follows from the observation that the fact that the order dependence between different proof rules is a major source of inefficiency in proof search. I consider this observation, from Wallen (1986), to be crucial. Deriving the same range of proof systems, each by a separate semantic argument, would make this point less clear.

This description of my methods does not apply to the final new result presented in this thesis, namely the extension of propositional proof systems in the case that the restriction theory does not satisfy the least Herbrand model property. This last result is proved semantically, with no further proof theoretical analysis. The success of the analysis of the earlier systems gives good ground for suggesting that such analysis should be applied to this more complicated case.

Section 1.4: Chapter by Chapter Outline

In Chapter 2 I introduce formal details of restricted quantification, including syntax and semantics as well as an extension of Smullyan's (1968) uniform notation to restricted quantification. I also discuss briefly how modal logics may be expressed by restricted quantification.

In Chapter 3 I introduce the framework within which I present proof systems throughout this thesis. This is D'Agostino's (1990) formalisation of ''expansion systems''. I choose this framework because many proof systems can be expressed naturally in it, as can the proof rules I present later to deal with restricted quantification. The archetypal expansion system is analytic tableau, proofs being branching trees and branches being terminated by explicit contradictions. However, resolution, for example, can be expressed easily as an expansion system. I also introduce Mondadori's (1989) tree structured proof system **KE**, which D'Agostino has shown to be of considerable interest because of its naturalness and advantageous proof theoretical properties.

In Chapter 3 I discuss various important properties of importance in expansion systems, such as soundness and completeness. In particular, a common argument in showing the completeness of tree based proof systems is that branches can be expanded *systematically* to ensure that no part of the search space is omitted. I show that this argument can be generalised to all expansion systems. Also, I formally distinguish between those expansion systems, like tableaux and **KE**, where a failed proof enables us to construct a countermodel, and expansion systems like resolution, where this is not so easy. I call the former ''countermodel'' expansion systems. This distinction is important because countermodel expansion systems can be extended for restriction quantification more easily than non-countermodel

systems.

In Chapter 4 I introduce a first extension of propositional expansion systems to restricted quantification. This is done by the introduction of new expansion rules for existential and universal quantifiers. Expansion of existentials introduces the restriction on the quantifier explicitly. The rules may only be applied if certain *side conditions* are satisfied at the point of expansion. The crucial side condition applies to the universal rule, and demands that the explicit restrictions on the branch force the restriction on the universal to be satisfied. This expansion system does not introduce any new branching structure into proofs. Furthermore, the restriction theory need only be considered when checking the side conditions. The advantage of this first proof system is that it is complete for a very wide range of restriction theories, namely for any theory satisfying a simple semantic condition, that each set of restrictions has a least Herbrand model. However, completeness only holds for countermodel expansion systems.

The fact that theory reasoning is incorporated solely into one side condition in the proof systems of Chapter 4 means that restricted quantification introduces no extra propositional reasoning more than unrestricted quantification. Also, the fact that the system is sound and complete in a very general case means that further study can identify particular subcases where refinements can be made. These properties of the systems of Chapter 4 make them extremely suitable for further study, as the next four chapters show. Much of this study concerns the elimination of order dependence of expansion rule application, arising from the side conditions in the proof systems.

In Chapter 5 I identify two important extra properties of restriction theories which allow the reduction of the order dependence of expansions in a proof. Whereas in Chapter 4, where the precise point in a tree where a universal expansion is applied is crucial to the satisfaction of the side condition, one can use a slightly weaker side condition, with the result that this order dependence is reduced. This weaker side condition relies only on global properties of a proof for its satisfaction. In Chapter 5 I identify a condition I call ''alphabetical monotonicity''. This ensures that the weaker side condition is still sound. Unfortunately, this soundness relies on the presence of another side condition which retains a lesser degree of order dependence. So I show that a stronger condition than alphabetical monotonicity, which I call ''non-vacuity'', means that this other side condition can be omitted while retaining soundness. Where non-vacuity holds, the only order dependence left in trees comes from the standard eigenvariable condition on existential expansions, that a name introduced from an existential formula must be a new name. This can be dealt with in ways well known from theorem proving in classical first order logic, as I discuss later in the thesis.

In Chapter 6 I give proofs of the satisfaction of alphabetical monotonicity and non-vacuity for various restriction theories, concentrating on modal and sorted logics. Non-vacuity is satisfied if the theory leads to serial or reflexive modal logics or ensures non-empty sorts in sorted logics.

In Chapter 7 I discuss one method that has been applied to classical logic for ensuring the satisfaction of the eigenvariable condition on existential instantiation. This is the use of reduction orderings due to Bibel (1987). The arguments I use are due to Wallen (1986, 1989). First I remove all side conditions from the proof systems, meaning that the expansion rules are used just to check for the existence of a closed tree. I have already established in Chapter 5 that the side condition using the restriction theory can be checked globally. Then, given a tree one can define a binary relation which, when acyclic, ensures that a tree satisfying the eigenvariable condition exists. Unfortunately, as I mentioned earlier, where alphabetical monotonicity holds but non- vacuity does not, a remnant of order dependence remains. This has to be catered for by using an unpleasant definition of closure: unpleasant because checking it involves more work than the usual definition. (The usual definition of closure is that identical formulas of opposite signs occur.) Where non-vacuity holds this usual definition is good enough. The work of this chapter generalises Wallen's (1989) approach to matrix characterisations of validity for modal logics. Unlike Wallen and Bibel, however, I do not discuss how matrices should be represented, or how to implement a theorem prover based on the characterisation of validity I derive in this chapter.

I showed earlier that where non-vacuity holds, no order dependence of expansion rules arises from the restriction theory. This means that the nature of propositional rules is irrelevant. The quantifier instantiation rules of Chapter 4 only applied to explicit quantifiers, and was only complete for countermodel expansion systems. In Chapter 8 I show that we can apply similar rules to quantifiers contained in complex formulas, so long as non-vacuity holds. That is, while before we could only instantiate explicit quantifiers, now we can instantiate soundly into formulas whose outermost connective is propositional. In this way any propositional proof system can be extended to restricted quantification, not just countermodel expansion systems.

All the work from Chapters 4 to 8 makes the assumption that the restriction theory satisfies the least Herbrand model property. In Chapter 9 I drop this assumption. I introduce a new rule for universal instantiation and show it to be sound and complete whether or not the least Herbrand model property holds. However, the new rule allows branching to occur due to universal instantiation, making it much less desirable than the rule from Chapter 4. I do not study the resulting proof systems in depth, as I did for the earlier proof systems from Chapter 5 to Chapter 8. Rather, given the work of those chapters, I pose the repetition of this work in the new context as a research problem. Solving this problem in a way that resulted in efficient proof systems would be of great importance to automated reasoning in logics such as temporal logics, whose accessibility theory fails the least Herbrand model property.

In Chapter 10 I consider work from the literature that is related to mine, as well as giving some conclusions and suggestions for further work.

Also included in this thesis is Appendix 1. Here, I discuss how the correctness of prefixed expansion systems like Fitting's prefixed tableaux (1983, Chapter 8) may be deduced from work presented in Chapters 4, 5, and 6 in this thesis.

Chapter 2

Restricted Quantification

In this chapter I introduce the syntax and semantics of logics of restricted quantification, logics which are the subject of study of this thesis. The basic semantic notion of restricted quantification is simple. Instead of usual, unrestricted, quantifiers read as "for all x" and "there is an x" we use restricted quantifiers, read as "for all x satisfying ρ" and "there is an x satisfying ρ" where ρ is a *restriction*. The question is, what is a restriction? Hailperin (1957), for example, allows arbitrary first order logic formulas. I will only allow instances of predicates to appear as restrictions.[1] As well as allowing quantification to be restricted, one can write down a theory concerning the restrictions. In this thesis I will demand that this theory be expressed in first order logic. I will call the theory a "restriction theory". The result is a language very much like Bürckert's (1990a) except that he considers arbitrary theories, possibly not expressible in first order logic. Taking this view of restricted quantification gives great generality, allowing sorted and modal logics to be expressed very easily.

In Section 2.1 I introduce formal details of restriction theories. In Section 2.2 I introduce, given a restriction theory Σ, a logic $\Sigma\mathcal{L}$ of restricted quantification with restrictions from the theory. In Section 2.3 I give formal semantics for unrestricted and restricted quantification. In Section 2.4 I extend Smullyan's (1968) uniform notation to deal with restricted quantification, a uniform notation I will use throughout the rest of this thesis. In Section 2.5 I discuss how modal logics may be expressed as logics of restricted quantification for appropriate restriction theories. In Section 2.6 I do the same for sorted logics.

Section 2.1: Restriction Theory Σ

I will introduce the notation of a Restriction Theory Σ. I assume that the language of Σ consists of some number of Σ-predicates

(*arity* 0) $\rho^0{}_1, \rho^0{}_2, \rho^0{}_3, \ldots$

(*arity* 1) $\rho^1{}_1, \rho^1{}_2, \rho^1{}_3, \ldots$

(*arity* 2) $\rho^2{}_1, \rho^2{}_2, \rho^2{}_3, \ldots$

etc.

and that terms in Σ are built from some collection of variables

$x, y, z, \ldots$

and some collections of Σ-functions, including constants regarded as functions of no arguments

(*arity* 0) $f^0{}_1, f^0{}_2, f^0{}_3, \ldots$

(*arity* 1) $f^1{}_1, f^1{}_2, f^1{}_3, \ldots$

(*arity* 2) $f^2{}_1, f^2{}_2, f^2{}_3, \ldots$

etc.

[1] As Felix Hovsepian very kindly pointed out to me, I do not need to demand this, at least not until Chapter 5. However, it makes presentation slightly easier.

Notation

I have every intention of abusing my notation as much as possible. For instance I will usually drop the arity superscript and often the disambiguating subscript, as well as using mnemonic names for predicates wherever helpful.

Definition 2.1.1 (Atom, Clause Syntax)

An *atom* is an instance of a predicate with each argument position instantiated with a term. For example if ρ^2 is a two-place predicate then $\rho^2(x, f(g(y,z)))$ would be an atom if f and g were functions and x, y, and z were variables.

A *clause* is an object of the form

$$B_1, B_2, \ldots B_m \rightarrow H_1, H_2, \ldots H_n$$

where either m or n may be 0, and each B_i, H_j, is a Σ-predicate instanced with suitable terms. Each B_i is said to be in the *body* of the clause and each H_j in the *head* of the clause.

The *empty* clause is the unique clause in which both m and n are 0.

A *goal* clause is a non-empty clause in which n is 0.

A *Horn* clause is a non-empty clause in which n is 0 or 1.

A *fact* is a Horn clause in which m is 0.

A *definite* clause is a clause in which n is 1 (that is, a non-goal Horn clause).

Definition 2.1.2 (Clause Semantics)

The empty clause has the value **f** in any model.

The semantics of a non-empty clause

$$B_1, B_2, \ldots B_m \rightarrow H_1, H_2, \ldots H_n$$

is given by the first order logic semantics (as defined in Section 2.3) of:

$$\forall x_1\ \forall x_2\ \cdots\ \forall x_k\ (B_1 \wedge B_2 \wedge \cdots \wedge B_m) \supset (H_1 \vee H_2 \vee \cdots \vee H_n)$$

where $x_1, x_2, \ldots, x_k$ are all the free variables in the clause.

Definition 2.1.3 (Restriction Theory Σ)

A restriction theory Σ is any set of clauses using only Σ-atoms, Σ-functions, and variables.

Section 2.2: Logic of Restricted Quantification - $\Sigma\mathcal{L}$

Throughout, I will use the propositional connectives ¬ (not), ∧ (and), ∨ (or), and ⊃ (material implication), together with the logical constants **t** (truth) and **f** (falsity).

As well as variables, constants, and functions as introduced in Section 2.1, I also assume that we have some countably infinite set of *names* available, including the distinguished name ε (which will play a special role later)

$$\varepsilon, a_1, a_2, a_3, \ldots$$

Names are distinct from constants because names will play a special role (in existential instantiation) in the proof systems in this thesis. To ensure names play their correct role I will assume that names are not used in the statement of the theory Σ, although constants might be. Where names are allowed, they may be used in terms in the same way as constants.

Given a restriction theory Σ, I can define a Σ-logic $\Sigma\mathcal{L}$. The predicates of $\Sigma\mathcal{L}$ are the predicates of Σ together with some $\mathcal{L}$-predicates:

(*arity* 0) $P^0{}_1, P^0{}_2, P^0{}_3, \ldots$

(*arity* 1) $P^1{}_1, P^1{}_2, P^1{}_3, \ldots$

(*arity* 2) $P^2{}_1, P^2{}_2, P^2{}_3, \ldots$

etc.

Terms in $\Sigma\mathcal{L}$ are built up from the same variables and functions as in Σ. It may seem odd that all functions in $\Sigma\mathcal{L}$ have to appear in Σ. However, the fact that all functions are officially Σ-functions does not mean that Σ has to say anything about them.

The crucial feature of $\Sigma\mathcal{L}$ is that Σ-literals are built into the syntax of $\Sigma\mathcal{L}$ in such a way that certain subformulas of a $\Sigma\mathcal{L}$-formula will be known to be Σ-formulas. This is achieved by introducing *restricted quantifiers* into the language of $\Sigma\mathcal{L}$.

A formula quantified by a restricted quantifier is a formula of the form

$$\forall x_{:\rho(x)}\ \phi(x)$$

or

$$\exists x_{:\rho(x)}\ \phi(x)$$

where $\rho(x)$ is a Σ-formula and $\phi(x)$ is some $\Sigma\mathcal{L}$ formula. I will call $\rho(x)$ the *restriction* of the quantification. I will also demand that Σ-atoms may *only* appear as restrictions.

Definition 2.2.1 (Syntax of a Σ-logic)

Given a restriction theory Σ and a set of $\mathcal{L}$-predicates, the well formed formulas of a Σ-logic are built as follows.

(i) any $\mathcal{L}$-atom is a well formed formula, as are the formulas **t** and **f**.

(ii) if ϕ and ψ are well formed formulas then so are $\phi\wedge\psi$, $\phi\vee\psi$, $\phi\supset\psi$, $\neg\phi$.

(iii) if ϕ is a well formed formula and ρ is a Σ-atom, then $\forall x_{:\rho}\ \phi$ and $\exists x_{:\rho}\ \phi$ are both well formed formulas.

Later I define the formal semantics for $\Sigma\mathcal{L}$. The semantics of the two restricted quantifiers will be exactly as if they were defined in terms of standard unrestricted quantifiers by this relativisation.

$$\forall x_{:\rho(x)}\ \phi(x) \ \equiv_{def}\ \forall x(\rho(x)\supset\phi(x))$$

and

$$\exists x_{:\rho(x)}\ \phi(x) \ \equiv_{def}\ \exists x(\rho(x)\wedge\phi(x))$$

Note that $\forall x_{:\rho(x)}$ and $\exists x_{:\rho(x)}$ are duals in the same way that $\forall$ and $\exists$ are duals, in that

$$\forall x_{:\rho(x)}\ \neg\phi(x) \ \equiv\ \neg\exists x_{:\rho(x)}\ \phi(x)$$

I demand that unrestricted quantifiers occur only in Σ. This will be convenient to simplify the presentation of proof systems in this thesis, and is no restriction to the expressiveness of $\Sigma\mathcal{L}$ since

$$\forall x\ \phi(x) \ \equiv\ \forall x_{:\mathbf{t}}\ \phi(x)$$

$$\exists x\ \phi(x) \ \equiv\ \exists x_{:\mathbf{t}}\ \phi(x)$$

Note that in general $\forall x_{:\rho(x)}\ \phi(x)$ does not imply $\exists x_{:\rho(x)}\ \phi(x)$, because it is possible that the universal quantification might be empty: there may be no x such that $\rho(x)$ holds. It might be that the restriction theory ensures that any restricted quantification is non-vacuous. Where this is so, better proof systems can be constructed than where vacuous quantification is possible. Indeed, investigation of this point is the main subject of Chapters 5 to 8 in this thesis.

Section 2.3: Semantics of Σ Logic

I have expressed the meaning of restricted quantification via a relativisation into classical first order logic. Casual readers need not concern themselves with a formal definition of the semantics of Σ-logic. However, I must include such a definition so that I can prove formally that the proof systems I give characterise the logic correctly.

The semantics I give are entirely standard, except for the changes required to incorporate Σ and restricted quantification. The substance of the following definitions is taken from Fitting (1990b, §5.3).

Definition 2.3.1 (Model)

A *model* **M** is a pair <**D**, **I**> where:

D is a non-empty set, called the *domain* of **M**;

I is a mapping, called an *interpretation* that associates:

to every name a some member $a^{\mathbf{I}} \in \mathbf{D}$;

to every n-place function symbol f some n-ary function

$$f^{\mathbf{I}} : \mathbf{D}^n \to \mathbf{D};$$

to every n-place relation symbol P some n-ary relation

$$P^{\mathbf{I}} \subseteq \mathbf{D}^n.$$

An *assignment* in a model **M** = <**D**, **I**> is a mapping **A** from the set of variables to the set **D**. We denote the image of the variable v under an assignment **A** by $v^{\mathbf{A}}$.

Given a model **M** = <**D**, **I**> and an assignment **A**, we associate each term t with a value $t^{\mathbf{I,A}}$ in **D** as follows:

1. for a name a, $a^{\mathbf{I,A}} = a^{\mathbf{I}}$;
2. for a variable v, $v^{\mathbf{I,A}} = v^{\mathbf{A}}$;
3. for a function symbol f,

$$(f(t_1, \ldots, t_n))^{\mathbf{I,A}} = f^{\mathbf{I}}(t_1^{\mathbf{I,A}}, \ldots, t_n^{\mathbf{I,A}}).$$

As well as the usual semantic definitions, and the definition of semantics of restricted quantification, I consider sets of formulas to be equivalent to the conjunction of all the formulas in the set.

Definition 2.3.2 (Semantics)

1 For the atomic cases, $(P(t_1, \ldots, t_n))^{\mathbf{I,A}} = \mathbf{t}$ if and only if $<t_1^{\mathbf{I,A}}, \ldots, t_n^{\mathbf{I,A}}> \in P^{\mathbf{I}}$, $\mathbf{t}^{\mathbf{I,A}} = \mathbf{t}$, and $\mathbf{f}^{\mathbf{I,A}} = \mathbf{f}$.

2 $(\neg X)^{\mathbf{I,A}} = \mathbf{t}$ if and only if $X^{\mathbf{I,A}} = \mathbf{f}$.

3 $(X \wedge Y)^{\mathbf{I,A}} = \mathbf{t}$ if and only if $X^{\mathbf{I,A}} = \mathbf{t}$ and $Y^{\mathbf{I,A}} = \mathbf{t}$.
$(X \vee Y)^{\mathbf{I,A}} = \mathbf{t}$ if and only if $X^{\mathbf{I,A}} = \mathbf{t}$ or $Y^{\mathbf{I,A}} = \mathbf{t}$.
$(X \supset Y)^{\mathbf{I,A}} = \mathbf{t}$ if and only if $X^{\mathbf{I,A}} = \mathbf{f}$ or $Y^{\mathbf{I,A}} = \mathbf{t}$.

4 $(\forall x\, \Phi(x))^{\mathbf{I,A}} = \mathbf{t}$ if and only if $\Phi(x)^{\mathbf{I,B}} = \mathbf{t}$ for every assignment **B** that is an x-variant of **A**.

5 $(\exists x\, \Phi(x))^{\mathbf{I,A}} = \mathbf{t}$ if and only if $\Phi(x)^{\mathbf{I,B}} = \mathbf{t}$ for some assignment **B** that is an x-variant of **A**.

6 $(\forall x_{:\rho(x)} \Phi(x))^{\mathbf{I,A}} = \mathbf{t}$ if and only if $\Phi(x)^{\mathbf{I,B}} = \mathbf{t}$ for every assignment **B** that is an x-variant of **A** and such that $\rho(x)^{\mathbf{I,B}} = \mathbf{t}$.

7 $(\exists x_{:\rho(x)} \Phi(x))^{\mathbf{I,A}} = \mathbf{t}$ if and only if $\Phi(x)^{\mathbf{I,B}} = \mathbf{t}$ for some assignment **B** that is an x-variant of **A** and such that $\rho(x)^{\mathbf{I,B}} = \mathbf{t}$.

8 If S is a set of formulas, then $S^{\mathbf{I,A}} = \mathbf{t}$ if and only if $X^{\mathbf{I,A}} = \mathbf{t}$ for each $X \in S$.

It is trivial to check that the relativisation of the previous section, together with (4) and (5) above, gives exactly the same semantics to restricted quantification as (6) and (7).

Definition 2.3.3 (Σ-model, Σ-satisfiable, Σ-entails, $\vDash_\Sigma$, Σ-valid)

A Σ-*model* is a model of Σ.

A formula Φ is *true in the model* **M** (or **M** *satisfies* Φ) iff $\Phi^{\mathbf{I,A}} = \mathbf{t}$ for all assignments **A**.

A formula Φ is Σ-*satisfiable* if and only if there is some Σ-model which satisfies ϕ. It is Σ-*unsatisfiable* if and only if it is not Σ-satisfiable.

If every Σ-model that satisfies a set of formulas $U = \{ \phi_1, \phi_2, \ldots \}$ also satisfies a formula ϕ, then U Σ-*entails* ϕ. This is written as

$$U \vDash_\Sigma \phi \quad or \quad \phi_1, \phi_2, \ldots \vDash_\Sigma \phi$$

A formula Φ is Σ-*valid* if and only if every Σ-model satisfies ϕ.

Section 2.4: The Uniform Notation for Restricted Quantification

In presenting the proof systems in this thesis, I shall use Smullyan's (1968) elegant idea of uniform notation, with a slight adaptation to suit my purposes. The insight behind the uniform notation is that sets of logical connectives behave in similar way. The only important distinctions are between those contexts where connectives that behave conjunctively, disjunctively, existentially, and universally. That is, in considering proof theory we only really have to consider four, schematic, connectives. All other details can be captured by listing how each logical connective fits into the schema.

The proof systems in this thesis will deal with *polarised* formulas, that is formulas associated with a polarity (or sign). It is to polarised formulas that the uniform notation applies.

Definition 2.4.1 (Polarity, Polarised Formula)

A *polarity* is a member of the set {**0**,**1**}.

A *polarised formula* is a formula associated with a polarity. For example the formula $P \wedge Q$ associated with polarity **0** would be written $\mathbf{0}P \wedge Q$.

For convenience, I will define the semantics of polarities. This will allow me to treat polarised formulas as if they were formulas of $\Sigma\mathcal{L}$, and to talk meaningfully about polarised formulas being satisfied or not in a model. Note that polarities do not become connectives by this move: they may only appear once, at the outside of a formula.

Definition 2.4.2 (Semantics of Polarised Formulas)

$(\mathbf{1}\,\Phi)^{\mathbf{I,A}} = \mathbf{t}$ if and only if $\Phi^{\mathbf{I,A}} = \mathbf{t}$.

$(\mathbf{0}\,\Phi)^{\mathbf{I,A}} = \mathbf{t}$ if and only if $\Phi^{\mathbf{I,A}} = \mathbf{f}$.

In the uniform notation, there are five types of polarised formulas: literals, α-type, β-type, γ-type and δ-type. You can think of an α-type formula as acting as an "and", a β-type formula as acting as an "or", a γ-type formula as acting as a universal restricted quantification, and a δ-type formula as acting as an existential restricted quantification.

Definition 2.4.3 (Literal)

Formulas containing no connectives are called *literals*. That is, a literal is a polarised formula **0**ϕ or **1**ϕ if ϕ is a $\mathcal{L}$-atom, or simply ρ if ρ is a Σ-atom.

Definition 2.4.4 (α-type formula)

α formulas act conjunctively. An α formula has two immediate subformulas α_1 and α_2. Those formulas of type α, and their subformulas, are shown below.

Formula α	Subformulas α_1	α_2
$\mathbf{1}\phi\wedge\psi$	$\mathbf{1}\phi$	$\mathbf{1}\psi$
$\mathbf{0}\phi\vee\psi$	$\mathbf{0}\phi$	$\mathbf{0}\psi$
$\mathbf{0}\phi\supset\psi$	$\mathbf{1}\phi$	$\mathbf{0}\psi$
$\mathbf{1}\neg\phi$	$\mathbf{0}\phi$	-

Definition 2.4.5 (β-type formula)

A β formula has two immediate subformulas, β_1 and β_2. The β-type formulas and their subformulas are defined below:

Formula β	Subformulas β_1	β_2
$\mathbf{0}\phi\wedge\psi$	$\mathbf{0}\phi$	$\mathbf{0}\psi$
$\mathbf{1}\phi\vee\psi$	$\mathbf{1}\phi$	$\mathbf{1}\psi$
$\mathbf{1}\phi\supset\psi$	$\mathbf{0}\phi$	$\mathbf{1}\psi$
$\mathbf{0}\neg\phi$	$\mathbf{1}\phi$	-

The issue of subformulas is more complicated for γ and δ formulas than for α and β-type formulas. This is because, in a subformula of, say, $\mathbf{1}\forall x_{:\rho(x)}\ \phi(x)$, the quantifier "$\forall x$" disappears and hence we need to instantiate the variable x in $\mathbf{1}\phi(x)$ by some name or term. For a particular term t, I shall write $\mathbf{1}\phi(t)$ for this. For each term t we must regard $\mathbf{1}\phi(t)$ as a subformula of the original quantification. This is represented in the following tables by indexing the subformula by the introduced terms.

Another point about subformulas of restricted quantifications is how to deal with the restrictions that arise. The main subformula of a γ or δ formula will be denoted by γ_0 or δ_0, and will be a polarised formula. The restriction of a γ or δ-type formula will be denoted by γ_r or δ_r. To emphasise that restrictions are different to normal formulas, they will be *un*polarised formulas. For instance, this means that expansion rules cannot be applied to any formula $\gamma_r(t)$ or $\delta_r(t)$. Certainly any reasoning that takes place concerning restrictions will have to be by some special purpose mechanism within a theory.

Notation

In this thesis, having arranged that subformulas of γ and δ-type formulas are indexed by a term, I shall often omit to mention the term where that term is unimportant. For instance, I might refer to γ_0 instead of $\gamma_0(t)$.

Definition 2.4.6 (γ-type formula)

γ formulas act as universal restricted quantifications. Formulas of type γ, and for a given term t their subformulas $\gamma_r(t)$ and $\gamma_0(t)$ are defined as follows:

Formula γ	Subformulas $\gamma_r(t)$	$\gamma_0(t)$
$\mathbf{1}\ \forall x_{:\rho(x)}\ \phi(x)$	$\rho(t)$	$\mathbf{1}\ \phi(t)$
$\mathbf{0}\ \exists x_{:\rho(x)}\ \phi(x)$	$\rho(t)$	$\mathbf{0}\ \phi(t)$

Definition 2.4.7 (δ-type formula)

δ formulas act as existential restricted quantifications. Formulas of type δ, and for a given term t their subformulas $\delta_r(t)$ and $\delta_0(t)$ are defined as follows:

Formula δ	Subformulas $\delta_r(t)$	$\delta_0(t)$
$\mathbf{1}\ \exists x_{:\rho(x)}\ \phi(x)$	$\rho(t)$	$\mathbf{1}\ \phi(t)$
$\mathbf{0}\ \forall x_{:\rho(x)}\ \phi(x)$	$\rho(t)$	$\mathbf{0}\ \phi(t)$

Section 2.5: Modal Logic as a Logic of Restricted Quantification

The primary motivation for my work for this thesis was the study of Lincoln Wallen's (1989) matrix proof systems for modal logics. In this thesis, it is convenient to regard modal logics as logics of restricted quantification, with the restrictions relating to the modal accessibility relation. To emphasise this convenience, I shall formally define modal logics in this way. Amongst others, Miura (1983) and Ohlbach (1988) have studied the expression of modal logics in this way, based on their Kripke semantics. I do not seek to introduce modal logics to readers unfamiliar with them. Such readers should consult an introduction to modal logics, such as Hughes and Cresswell (1968), Chellas (1980), or Simmons (1992).

In this thesis I do not consider *non-normal* modal logics. Non-normal modal logics are those in which the set of formulas true in a world is not forced to be consistent. Most modal logics seen in the literature of Artificial Intelligence and Computer Science are normal modal logics. Fitting (1983) discusses theorem proving methods for non-normal modal logics.

Section 2.5.1: Propositional Modal Logic

We can translate a formula of modal logic to Σ-logic using the translation function τ_m of one argument.

$$\tau_m(\phi) = \tau_m(\phi,\varepsilon)$$

where ε is an arbitrary name for a starting world, and the translation function τ_m of two arguments is defined in the obvious way for propositional connectives and as follows for □ and ◇:

$$\tau_m(\Box\phi,w) = \forall x_{:w\mathcal{R}x}\ \tau_m(\phi,x)$$

$$\tau_m(\Diamond\phi,w) = \exists x_{:w\mathcal{R}x}\ \tau_m(\phi,x)$$

Various conditions on the accessibility relation $\mathcal{R}$ are defined as follows:

Theory	Clause
Seriality	$\rightarrow w\mathcal{R}next(w)$
Reflexivity	$\rightarrow w\mathcal{R}w$
Symmetry	$w\mathcal{R}x \rightarrow x\mathcal{R}w$
Transitivity	$(w\mathcal{R}x \wedge x\mathcal{R}y) \rightarrow w\mathcal{R}y$

In the clause for seriality, the function "*next*" is a skolem function. So for any world w, $next(w)$ is some world accessible from w. Note that this is not the same as the interpretation of *next* in some temporal logics, where $next(w)$ refers to the unique successor of w.

Section 2.5.2: First Order Modal Logic

Propositional modal logics may be extended to first order logics, but there is not just one first order extension of each modal logic. The problem is that we must decide how the universe of individuals varies from possible world to possible world. Different first order variants of modal logics are appropriate for different applications. There are four common variants of first order modal logic, as follows.

- constant domain, in which the set of individuals is assumed to be the same in each possible world
- varying domain, in which we make no assumptions on how the sets of individuals in different worlds are related
- cumulative domain, in which any individual existing in a world is assumed to exist anywhere which is accessible from that world
- descending domain, in which any individual existing in a world is assumed to exist anywhere that world is accessible from.

The choice of first order variant can be made independently of the choice of propositional modal logic. The accessibility theory can interact with the theory of the domains, though. For example, if the accessibility theory includes symmetry then the cumulative domain and the descending domain variants are identical.

The effect of these different variants can be seen by considering the *Barcan formula* and its converse. The Barcan formula (for the predicate P) is

$$\forall x \, \Box P(x) \supset \Box \forall x \, P(x).$$

This formula is true in the descending and constant domain variants of first order modal logic.

The converse of the Barcan formula (for the predicate P) is

$$\Box \forall x \, P(x) \supset \forall x \, \Box P(x).$$

This formula is true in the cumulative and constant domain variants of first order modal logic.

Different translations into Σ-logic are used for constant domain modal logics and for the rest.

For constant domain modal logic, the translation τ_m is extended by:

$$\tau_m^c(\forall x \, \phi, w) = \forall x_{:dom(x)} \, \tau_m^c(\phi, w)$$

$$\tau_m^c(\exists x \, \phi, w) = \exists x_{:dom(x)} \, \tau_m^c(\phi, w)$$

No change is made to the theory Σ. The restriction "$dom(x)$", read as "x is in the domain" provides an implicit two sorted structure onto the set of terms: those that represent possible worlds and those that represent individuals in the (constant) domain.

For the other variants, life is a little more complicated. We need to add a new Σ-predicate *in*, where $x \; in \; w$ is read as "the object x exists in the domain of world w". The translation τ_m is extended by:

$$\tau_m^v(\forall x \, \phi, w) = \forall x_{:x \, in \, w} \, \tau_m^v(\phi, w)$$

$$\tau_m^v(\exists x \, \phi, w) = \exists x_{:x \, in \, w} \, \tau_m^v(\phi, w)$$

The following additions are made to Σ:

Variant	Clause
Varying Domain	(none)
Cumulative Domain	$x \; in \; w_1, \; w_1 \mathcal{R} w_2 \rightarrow x \; in \; w_2$
Descending Domain	$x \; in \; w_2, \; w_1 \mathcal{R} w_2 \rightarrow x \; in \; w_1$

Section 2.6: Sorted Logics as Logics of Restricted Quantification

Sorted logics can be expressed as logics of restricted quantification even more easily than modal logics. Again, I do not seek to introduce sorted logics: Cohn (1989a) provides an introductory survey of sorted logics.

In simple sorted logics, the only syntactic extension to predicate logic is the attachment of *sorts* to variables. The change to the semantics of the logics arises from splitting the domain of discourse into a

set of sorts, so that each individual belongs to only one sort. No relationships between the various sorts exist. Such an approach is described, for example, in Enderton (1972). For example, one might write "every man loves a woman" as

$$\forall x_{man}\ \exists y_{woman}\ loves(x,y)$$

Such a logic can easily be relativised into a Σ-logic. The above example would become

$$\forall x_{:man(x)}\ \exists y_{:woman(x)}\ loves(x,y)$$

Since in the simple sorted logics, nothing can be said about the relationship between sorts, it may seem that Σ must be empty. However, one decision we must make is whether or not each sort should have non-empty domains. If we do not make any demand on the domain of any sorts then indeed the theory Σ will be empty. If we want to ensure that each sort is non-empty, then we must add clauses to Σ ensuring this. For men and women we might write:

$$\rightarrow man(adam)$$

$$\rightarrow woman(elizabeth)$$

As a simple extension of sorted logic one can impose a partial ordering on sorts, for instance by declaring:

$human \sqsupset man$
$human \sqsupset woman$
$woman \sqsupset spinster$
$man \sqsupset bachelor$

This partial ordering, as usual, is assumed to be transitive. If the relation $sort_1 \sqsupset sort_2$ holds this is interpreted as demanding that every individual of $sort_2$ is also an individual of $sort_1$. This extension of sorted logic is called *order sorted* logic. Order sorted logic is discussed at length by Schmidt-Schauss (1989). With the above orderings, one could express "everybody hates somebody" as

$$\forall x_{human}\ \exists y_{human}\ hates(x,y)$$

while with only the sorts *man* and *woman* available we would have to write:

$[\forall x_{man}\ (\exists y_{man}\ hates(x,y) \vee \exists z_{woman}\ hates(x,z))]$
$\wedge\ [\forall x_{woman}\ (\exists y_{man}\ hates(x,y) \vee \exists z_{woman}\ hates(x,z))]$.

Order sorted logics can easily be expressed as Σ-logics, the above examples giving rise to the following clauses:

$$man(x) \rightarrow human(x)$$

$$woman(x) \rightarrow human(x)$$

$$spinster(x) \rightarrow woman(x)$$

$$bachelor(x) \rightarrow man(x)$$

The transitivity of $\sqsupset$ is mirrored by the transitivity of implication.

In order sorted logics, typically functions are also sorted. This means that as well as sort ordering declarations, we must have function declarations. For example, we might declare the functions *fiance*, *fiancee*, and *best_man* as follows:

$fiance : spinster \rightarrow bachelor$
$fiancee : bachelor \rightarrow spinster$
$best\_man : fiance \times fiancee \rightarrow man$

These function declarations can again be expressed very simply as clauses in Σ. They would be written:

$$spinster(x) \rightarrow bachelor(fiance(x))$$
$$bachelor(x) \rightarrow spinster(fiancee(x))$$
$$fiance(x) \wedge fiancee(y) \rightarrow man(best\_man(x,y))$$

More generally, a declaration

$$f : sort_1 \times sort_2 \times \cdots \times sort_n \rightarrow sort_0$$

would be translated as

$$sort_1(x_1) \wedge sort_2(x_2) \wedge \cdots \wedge sort_n(x_n) \rightarrow sort_0(f(x_1, x_2, \ldots, x_n))$$

As in simple sorted logics, we have to add suitable clauses if we wish to ensure that each sort is non-empty. Such clauses will simply be of the form:

$$\rightarrow sort_i(t)$$

where t is some term built up from Σ-constants and Σ-functions.

In order sorted logics, polymorphism or overloading of function symbols is not allowed. That is, each function may only be declared once. Also, the sort heirarchy declared by $\sqsupset$ must be a tree. Cohn (1987) introduced a very expressive polymorphic sorted logic in which the sort heirarchy may be a lattice. Such a logic cannot easily be modelled as a logic of restricted quantification.

Chapter 3

Expansion Systems

In this chapter I introduce a simple way of avoiding talking about specific proof systems. I follow D'Agostino's (1990) definition of ''expansion systems'', which captures the idea of a tree based proof system in which we search for contradictions. Tableau, resolution, and Mondadori's (1989) **KE** can all easily be expressed as expansion systems, although resolution does not use the tree structure inherent in expansion systems. In the rest of this thesis, I show how to extend certain expansion systems under certain circumstances to deal with restricted quantification. I avoid explicitly choosing a single system, thereby giving general results.

To the reader who wishes to skim this chapter:

> *Until Chapter 8 I only deal with ''countermodel expansion systems''. These are just like tableaux in almost all respects. You may wish to glance at Section 3.5 where I define "positions", a concept I use throughout this thesis.*

In Section 3.1 I introduce expansion systems formally. I also define ''countermodel'' expansion systems. The characteristic of a countermodel expansion tree is that an open branch easily defines a model for the formulas on the branch. I also give examples of propositional expansion systems and examples of proofs in those systems. In Section 3.2 I show how the common argument of systematically searching trees for an open branch (see for instance Smullyan 1968) can be given quite generally for any expansion system. In Section 3.3 I discuss soundness and completeness of expansion systems, showing how the argument of Section 3.2 helps to prove completeness. I also prove soundness and completeness for the propositional systems introduced in Section 3.1. In Section 3.4 I show how König's Lemma can be used to prove that if there is a proof there must be a finite proof in an expansion system.

In Section 3.5 I define the notion of "positions" in expansion trees. Usually in this thesis I will discuss positions instead of formulas when discussing proof systems. This is because positions are entities that contain all the structural information that a formula does, and more besides. A position contains historical information about how formulas arise in expansion trees, information that can be different for two different occurrences of the same formula. This information will often be important.

Section 3.1: Definition of Expansion Systems

The following definitions are taken, with slight changes to fit my syntax, from D'Agostino (1990, §2.6.1).

Definition 3.1.1 (Expansion rule, Expansion system, Saturated)

1. An $n \times m$ *expansion rule* R is a relation between n-tuples of polarised formulas and m-tuples of polarised formulas, with $n \geq 0$ and $m \geq 1$. Expansion rules may be represented as follows:

$$\frac{\begin{array}{c}\chi_1 \\ \vdots \\ \chi_n\end{array}}{\nu_1 \mid \cdots \mid \nu_m}$$

   where the χ_i's and the ν_i's are schemes of polarised formulas.

   We say that the rule has n premises and m conclusions. If $m = 1$ we say that the rule is of *linear type*, otherwise we say that the rule is of *branching type*.

2. An *expansion system* **S** is a finite set of expansion rules.
3. We say that S,X is an *expansion* of S under an $n \times m$ rule R if there is an n-tuple a of elements of S such that X belongs to some m-tuple b in the set of images of a under R.
4. Let R be an $n \times m$ expansion rule. A set U is *saturated under R* or *R-saturated* if for every n-tuple a of elements of U and every m-tuple b in the set of images of a under R, at least one element of b is also in U.
5. A set U is **S**-*saturated* if it is R-saturated for every rule R of **S**.

Of course the expansion rules are read as tree construction rules. This is defined formally by:

Definition 3.1.2 (S-tree)

Given an expansion system **S** and a set of formulas S, we can build trees as follows.

1. The empty tree is an **S**-tree for S.
2. Given any branch λ in an **S**-tree for S and any formula $X \in S$, the tree in which a node labelled by X is added to λ is an **S**-tree for S.
3. Given any branch λ in an **S**-tree for S, any formulas $X_1, \ldots, X_n$ on the branch, and an expansion rule R such that $(Y_1, \ldots, Y_m)$ is the image under R of $(X_1, \ldots, X_n)$, the tree in which λ is extended by m separate nodes, each node i labelled by Y_i, is an **S**-tree for S.

Definition 3.1.3 (closed, open, complete)

Let λ be a branch of an **S**-tree.

We say that λ is *closed* if the set of its nodes is explicitly inconsistent, that is if it contains two signed formulas $\mathbf{0}X$ and $\mathbf{1}X$ for some formula X. Otherwise it is *open*.

A tree $\mathcal{T}$ is *closed* if all its branches are closed, and *open* otherwise.

We also say that a branch λ is *complete* if it is closed or the set of its nodes is **S**-saturated.

A tree $\mathcal{T}$ is *completed* if all its branches are complete.

D'Agostino says (1990, §2.6.2):

> ''Any set of rules meeting our definition of analytic refutation system can be considered an adequate formalisation of the idea of proving validity by a failed attempt to construct a countermodel.''

For my purposes, this is not quite an adequate formalisation. Nothing ensures that a countermodel can be found from a completed tree. Although many expansion systems can ensure that a countermodel exists from an open completed tree, some do not explicitly give a countermodel. This is the crucial property I will need. The following definition expresses this property more formally.

Definition 3.1.4 (Countermodel Expansion System)

An expansion system **S** is a *countermodel expansion system* if for any **S**-saturated set of formulas U which is not explicitly inconsistent:

there is at least one model satisfying each atomic formula in U; and

any model **M** satisfies each formula in U if **M** satisfies each atomic formula in U.

That is to say, if **S** is a countermodel expansion system, then we can find a model for U by finding any model **M** such that for any atomic formula P:

if $\mathbf{1}P \in U$ then **M** satisfies P;

if $\mathbf{0}P \in U$ then **M** does not satisfy P.

Note that resolution is not a countermodel expansion system. For example, suppose U consists of the single clause $\mathbf{1}P \vee Q$. This is resolution-saturated. As there are no signed propositions in U, any propositional model satisfies the conditions above. Unfortunately such models include those which fail to satisfy $P \vee Q$.

Because my work involves extending propositional systems to first order systems, I need to ensure that propositional systems leave quantifiers alone. Then I can know that the quantifier rules I will introduce will act as I intend them to, without interference from the propositional rules. To this end I make the following definition.

Definition 3.1.5 (Propositional Expansion System)

> An expansion system **S** is a *propositional expansion system* if it treats each quantification exactly as if it were a proposition.

This definition ensures that propositional expansion systems cannot use the structure of quantifications. Furthermore, anything that the system can do to a proposition it can do to a quantification. For example, if it can introduce $\mathbf{1}P$ from $\mathbf{0}\neg P$ then it must be able to introduce $\mathbf{1}\forall x\, \Phi$ from $\mathbf{0}\neg\forall x\, \Phi$. All the expansion systems for propositional logic that I consider in this thesis will certainly be propositional expansion systems.

Section 3.1.1: Running Example for Propositional Expansion Systems

To illustrate the various expansion systems that I will consider, I will use the same example and give proofs in the different systems. The formula is:

$$(P \equiv Q) \wedge (P \equiv \neg Q)$$

It is not difficult to see that the formula is unsatisfiable. Suppose P is true. Then from the first conjunct Q must be true. But if Q is true then from the second conjunct P must be false. So P cannot be true. But if P is false then from the first conjunct Q must be false and from the second conjunct P must be true. So P can be neither true nor false and the formula is unsatisfiable.

This formula is not acceptable to proof systems which cannot deal with equivalence. In that case we need an equivalent formula which does not use equivalence. I will use the following equivalent formula.

$$(P \vee Q) \wedge (P \vee \neg Q) \wedge (\neg P \vee Q) \wedge (\neg P \vee \neg Q)$$

Section 3.1.2: An example of an expansion system: Analytic Tableau

Perhaps the most standard and familiar example of an expansion system is analytic tableau. I will call the system **Tab**. I present **Tab** using the uniform notation to make the presentation as simple as possible. There is one expansion rule for each type of polarised formula - α- and β-types.

Expansion Rules for **Tab**

(α)	$\dfrac{\alpha}{\begin{array}{c}\alpha_1\\ \alpha_2\end{array}}$	(β)	$\dfrac{\beta}{\beta_1 \mid \beta_2}$

Running example: Tableau without equivalence

$$
\begin{array}{c}
\mathbf{1}\,(P \vee Q) \\
\mathbf{1}\,(P \vee \neg Q) \\
\mathbf{1}\,(\neg P \vee Q) \\
\mathbf{1}\,(\neg P \vee \neg Q) \\
\hline
\begin{array}{cc}
\mathbf{1}\,P & \mathbf{1}\,Q \\
\hline
\begin{array}{c}
\mathbf{1}\,\neg P \\ \hline \mathbf{0}\,P \\ \hline \otimes
\end{array}
&
\begin{array}{c}
\mathbf{1}\,Q \\ \hline
\begin{array}{cc}
\begin{array}{c}\mathbf{1}\,\neg P \\ \hline \mathbf{0}\,P \\ \hline \otimes\end{array}
&
\begin{array}{c}\mathbf{1}\,\neg Q \\ \hline \mathbf{0}\,Q \\ \hline \otimes\end{array}
\end{array}
\end{array}
\end{array}
\end{array}
\qquad
\begin{array}{cc}
\mathbf{1}\,Q & \\
\hline
\begin{array}{c}
\mathbf{1}\,P \\ \hline
\begin{array}{cc}
\begin{array}{c}\mathbf{1}\,\neg P \\ \hline \mathbf{0}\,P \\ \hline \otimes\end{array}
&
\begin{array}{c}\mathbf{1}\,\neg Q \\ \hline \mathbf{0}\,Q \\ \hline \otimes\end{array}
\end{array}
\end{array}
&
\begin{array}{c}\mathbf{1}\,\neg Q \\ \hline \mathbf{0}\,Q \\ \hline \otimes\end{array}
\end{array}
$$

Notation

> *I use the symbol "⊗" simply to give an explicit indication that the branch is closed. When giving examples of expansion trees, such as the one above, I omit the vertical line separating branches, preferring instead to use the clear spatial separation between branches.*

It is interesting to note that the tableau above ends in six branches, yet there are only two propositions in the example. Indeed, my running example is an instance of a general class of formulas which prove an unexpected result due to D'Agostino. Analytic tableaux have usually been thought to be uniformly better than truth tables, but for this class of examples there is no polynomial bounding the size of the tableau in terms of the size of the truth table! D'Agostino conjectured this result in his D.Phil. thesis (1990) and has since proved it (D'Agostino, personal communication, June 1991).

Section 3.1.3: An example of an expansion system: KE

Recently, another tableau-like system has been shown to be of considerable interest. This is Mondadori's (1989) **KE**.

I present **KE** using the uniform notation to make the presentation as simple as possible.

Notation

> *In the following presentation, I use X^c to mean the complement of X. That is, if X is $\mathbf{1}P$ then X^c is $\mathbf{0}P$ and if X is $\mathbf{0}P$ then X^c is $\mathbf{1}P$.*

Expansion Rules for **KE**

(A)	$\dfrac{\alpha}{\begin{array}{c}\alpha_1\\ \alpha_2\end{array}}$		
(B1)	$\dfrac{\begin{array}{c}\beta\\ \beta_1{}^c\end{array}}{\beta_2}$	(B2)	$\dfrac{\begin{array}{c}\beta\\ \beta_2{}^c\end{array}}{\beta_1}$
(PB)	$\dfrac{}{X \quad X^c}$		

Running example: KE without equivalence

$$
\begin{array}{c}
\mathbf{1}\,(P \vee Q) \\
\mathbf{1}\,(P \vee \neg Q) \\
\mathbf{1}\,(\neg P \vee Q) \\
\mathbf{1}\,(\neg P \vee \neg Q) \\
\hline
\begin{array}{c}
\mathbf{1}\,P \\ \hline \mathbf{1}\,Q \\ \hline \mathbf{1}\,\neg Q \\ \hline \mathbf{0}\,Q \\ \hline \otimes
\end{array}
\qquad
\begin{array}{c}
\mathbf{0}\,P \\ \hline \mathbf{1}\,\neg Q \\ \hline \mathbf{0}\,Q \\ \hline \mathbf{1}\,P \\ \hline \otimes
\end{array}
\end{array}
$$

As well as Mondadori, **KE** has been studied extensively by D'Agostino (1990). D'Agostino shows, for example, that there are classes of examples for which tableaux cannot polynomially simulate the **KE** proof, and also argues that any tableaux search method can be copied by **KE** with in the worst case a linear overhead but in the best case a great improvement in efficiency. Goré and D'Agostino (1991) have compared simple implementations of tableaux and **KE**, with results tending to confirm this.

Section 3.1.4: Extending Tab for equivalence

The (α) and (β) rules in **Tab** do not apply to formulas whose top connective is ≡ (equivalence). This is rather unfortunate. Without rules for equivalence we must translate any formula with equivalence to one without it. This translation can increase the size of a formula exponentially. It is not difficult to come up with tableau rules for equivalence. Indeed the following simple rules work. There is no particular point in introducing some more uniform notation: it would be as complex as the notation it replaces.

Tableau Expansion Rules for Equivalence

(**1**≡)	$\mathbf{1}\,X \equiv Y$	
	$\mathbf{1}X$	$\mathbf{0}X$
	$\mathbf{1}Y$	$\mathbf{0}Y$
(**0**≡)	$\mathbf{0}\,X \equiv Y$	
	$\mathbf{1}X$	$\mathbf{0}X$
	$\mathbf{0}Y$	$\mathbf{1}Y$

Running example: Tableau with equivalence

$$
\begin{array}{c}
\mathbf{1}\,P \equiv Q \\
\mathbf{1}\,P \equiv \neg Q \\
\hline
\begin{array}{c}
\mathbf{1}\,P \\
\mathbf{1}\,Q \\
\hline
\begin{array}{cc}
\begin{array}{c}
\mathbf{1}\,P \\
\mathbf{1}\,\neg Q \\
\hline
\mathbf{0}\,Q \\
\hline
\otimes
\end{array}
&
\begin{array}{c}
\mathbf{0}\,P \\
\mathbf{0}\,\neg Q \\
\hline
\otimes
\end{array}
\end{array}
\end{array}
&
\begin{array}{c}
\mathbf{0}\,P \\
\mathbf{0}\,Q \\
\hline
\begin{array}{cc}
\begin{array}{c}
\mathbf{1}\,P \\
\mathbf{1}\,\neg Q \\
\hline
\otimes
\end{array}
&
\begin{array}{c}
\mathbf{0}\,P \\
\mathbf{0}\,\neg Q \\
\hline
\mathbf{1}\,Q \\
\hline
\otimes
\end{array}
\end{array}
\end{array}
\end{array}
$$

Section 3.1.5: Extending KE for equivalence

We have just seen that tableaux can be extended easily to cover equivalence: easily, but perhaps not naturally since the new rules do not look like the old ones. On the other hand, **KE** can be extended to equivalence by introducing rules just like the old ones.[1]

[1] Although the material in this section is easy, it is new, as far as I know.

KE Expansion Rules for Equivalence

(**1**E+L)	$\begin{array}{c}\mathbf{1}\,X \equiv Y\\ \mathbf{1}X\\ \hline \mathbf{1}Y\end{array}$	(**1**E-L)	$\begin{array}{c}\mathbf{1}\,X \equiv Y\\ \mathbf{0}X\\ \hline \mathbf{0}Y\end{array}$
(**0**E+L)	$\begin{array}{c}\mathbf{0}\,X \equiv Y\\ \mathbf{1}X\\ \hline \mathbf{0}Y\end{array}$	(**0**E-L)	$\begin{array}{c}\mathbf{0}\,X \equiv Y\\ \mathbf{0}X\\ \hline \mathbf{1}Y\end{array}$
(**1**E+R)	$\begin{array}{c}\mathbf{1}\,X \equiv Y\\ \mathbf{1}Y\\ \hline \mathbf{1}X\end{array}$	(**1**E-R)	$\begin{array}{c}\mathbf{1}\,X \equiv Y\\ \mathbf{0}Y\\ \hline \mathbf{0}X\end{array}$
(**0**E+R)	$\begin{array}{c}\mathbf{0}\,X \equiv Y\\ \mathbf{1}Y\\ \hline \mathbf{0}X\end{array}$	(**0**E-R)	$\begin{array}{c}\mathbf{0}\,X \equiv Y\\ \mathbf{0}Y\\ \hline \mathbf{1}X\end{array}$

Although it is preferable to include them for symmetry, the right handed rules are unnecessary in the sense that they can be derived by an application of (PB) and one of the left handed rules. For example, (**0**E+R) could be derived as follows with an application of (**0**E+L) on the right hand branch:

$$\begin{array}{c}\mathbf{0}\,X \equiv Y\\ \mathbf{1}Y\\ \hline \mathbf{0}X \quad \begin{array}{c}\mathbf{1}X\\ \hline \mathbf{0}Y\\ \hline \otimes\end{array}\end{array}$$

Running example: KE with equivalence

$$\begin{array}{c}\mathbf{1}\,P \equiv Q\\ \mathbf{1}\,P \equiv \neg Q\\ \hline \begin{array}{c}\mathbf{1}\,P\\ \hline \mathbf{1}\,Q\\ \hline \mathbf{1}\,\neg Q\\ \hline \mathbf{0}\,Q\\ \hline \otimes\end{array} \qquad \begin{array}{c}\mathbf{0}\,P\\ \hline \mathbf{0}\,Q\\ \hline \mathbf{1}\,P\\ \hline \otimes\end{array}\end{array}$$

This expansion tree is identical to the tree produced in Section 3.1.3 for **KE** without completeness, except that one expansion of a negation in the earlier tree is not needed here. It says something about the naturalness of **KE**, and the new rules for equivalence, that the quite drastic translation into clause form did not affect the proof at all. The extension of tableaux for equivalence does not have this property: the trees given in Sections 3.1.2 and 3.1.4 are quite different from each other.

Section 3.2: Systematic Construction of Completed Trees

Usually, the first step in proving the semantic completeness of an expansion system is to give a systematic procedure for constructing a completed tree. In fact, this argument can be generalised to apply to any expansion system. The only restriction is that the set of well formed formulas must be countable. This certainly holds in first order logics as long as the set of names, function symbols, and constants is countable.

Theorem 3.2.1: Completed Tree Existence Theorem

If the set of well formed formulas is countable, then in any expansion system **S**, *there is a completed* **S**-*tree for any finite set of formulas S.*

Proof

Consider any set of formulas U. Define a *potential expansion* from U to be an m-tuple $(Y_1, \ldots, Y_m)$ such that there is an n-tuple $(X_1, \ldots, X_n)$ of elements of U of which $(Y_1, \ldots, Y_m)$ is the image under some **S**-expansion rule. Given any set of formulas, the set of potential expansions from U is countable, since the set of well formed formulas is countable, and the set of all finite tuples of a countable set is countable. Hence we can number the potential expansions with positive integers.

I now define the systematic procedure for building a tree.

A **S**-tree $\mathcal{T}$ for a finite set of formulas S is *systematic* if it satisfies the following conditions:

the first formulas in $\mathcal{T}$ are the formulas in S;

if a node in $\mathcal{T}$ is the endnode of an incomplete branch λ containing a set of formulas U, then the following potential expansion is applied to λ in $\mathcal{T}$: the least numbered potential expansion from U all of whose m formulas do not appear in U.

Now, given any finite set S of formulas, consider a systematic tree $\mathcal{T}$ for S. To show it is completed, I show that any branch λ_∞ is complete. The subscript "∞" indicates that λ_∞ may be an infinite branch.

If λ_∞ is closed, then it is by definition complete.

Suppose that λ_∞ is open. Then λ_∞ consists of a number of finite subbranches $\lambda_1, \lambda_2, \ldots$ and associated sets of formulas $U_1, U_2, \ldots$ Note that $U_1 \subseteq U_2 \subseteq \cdots$ Then $U_\infty = \bigcup_{i=1}^{\infty} U_i$.

I must show that U_∞ is **S**-saturated. It is enough to show that for any potential expansion $(Y_1, \ldots Y_m)$ from U_∞, at least one Y_i is in the set U_∞.

As a potential expansion from U_∞, $(Y_1, \ldots, Y_m)$ must be the image of some n-tuple $(X_1, \ldots, X_n)$ of elements of U_∞. Then there must be some p such that each X_i is in the set U_p. So $(Y_1, \ldots, Y_m)$ is a potential expansion from U_p, associated with some finite number q. Certainly it remains a potential expansion from $U_p, U_{p+1}, \ldots U_{p+q}$. By the pigeonhole principle,[2] the number associated with at least one of the $q+1$ expansions applied to $U_p, U_{p+1}, \ldots U_{p+q}$ must be greater than q. Say it was the expansion applied to U_j. Since the tree $\mathcal{T}$ is systematic, some Y_i must appear in U_j. So $Y_i \in U_\infty$. This is exactly what I needed to show. Hence U_∞ is **S**-saturated and the branch λ_∞ is complete.

□

Section 3.3: Soundness and Completeness of Expansion Systems

Definition 3.3.1 (Soundness, Completeness)

An $n \times m$ expansion rule R is *sound* iff, for any satisfiable set S containing an n-tuple $(X_1, \ldots, X_n)$ that yields an m-tuple $(Y_1, \ldots, Y_m)$ under R, then there is some Y_i such that S, Y_i is satisfiable.

An expansion system **S** is *sound* iff

a set of signed formulas S is unsatisfiable if there is a closed **S**-tree for S.

An expansion system **S** is *complete* iff

there is a closed **S**-tree for a set of signed formulas S if S is unsatisfiable.

The simple relationship between sound expansion *rules* and *systems* is made clear by the following observation.

[2] that $n+1$ pigeons cannot fit in to n holes.

Observation 3.3.2

If all the rules in an expansion system are sound, then that expansion system is sound. Furthermore, if a sound expansion system is complete then all the expansion rules are sound.

Proof

The first claim can be checked by a simple induction argument on the size of expansion trees, using the soundness of each expansion rule application.

Now suppose that **S** is a complete expansion system containing an unsound $n \times m$ expansion rule such that $X_1, \ldots, X_n$ is satisfiable but that for each i, $X_1, \ldots, X_n, Y_i$ is unsatisfiable. Then, from the completeness of **S**, there are closed **S**-trees for each set $X_1, \ldots, X_n, Y_i$. Combining these trees yields a closed **S**-tree for the satisfiable set
$X_1, \ldots, X_n$, showing that **S** is unsound.

□

Having the general Completed Tree Existence Theorem (3.2.1) greatly simplifies proving that a particular expansion system is complete. In fact, any countermodel expansion system is complete for finite sets of formulas, as the following theorem shows.

Theorem 3.3.3: Completeness of Countermodel Expansion Systems

If **S** *is a countermodel expansion system, and a finite set S of formulas is unsatisfiable, then there is a closed* **S**-*tree for S.*

Proof

I show the contrapositive of the theorem: if there is no closed **S**-tree for S then S is satisfiable.

By the Completed Tree Existence Theorem, there is a completed tree for S. By assumption, this tree must be open and so contain an open branch λ. The set of formulas U on λ is **S**-saturated and not explicitly inconsistent. Since **S** is a countermodel expansion system, there is a model which satisfies each formula in U. Since $S \subseteq U$ the model satisfies each formula in S.

□

In the rest of this section I will justify the soundness and completeness of the particular expansion systems I introduced in Section 3.1, using this Theorem for completeness. All these proofs are straightforward. I omit the proofs for analytic tableaux, giving only proofs for the less familiar **KE**. Proof of the following theorem can be found in sources such as Smullyan (1968).

Theorem 3.3.4: Soundness and Completeness of Tab

Tab *is a sound and complete expansion system for propositional logic if the equivalence symbol does not appear.*

□

Showing that the extension of **Tab** for equivalence is correct is entirely straightforward.

Theorem 3.3.5: Soundness and Completeness of Tab with equivalence

Tab *extended for equivalence is a sound and complete for propositional logic.*

□

Theorem 3.3.6: Soundness of KE

KE *is a sound expansion system for propositional logic.*

Proof

I must show that all the expansion rules in **KE** are sound in propositional logic. The soundness of the rules in fact follows easily from the semantics of propositional logics.

(A) Suppose a model satisfies α. Then it must also satisfy α_1 and α_2.

(B1) Suppose a model satisfies β and $\beta_1{}^c$. Since it satisfies β it must satisfy either β_1 or β_2. It certainly doesn't satisfy β_1 so the model satisfies β_2.

(B2) Suppose a model satisfies β and $\beta_2{}^c$. Since it satisfies β it must satisfy either β_1 or β_2. It certainly doesn't satisfy β_2 so the model satisfies β_1.

(PB) Given any formula X, any model satisfies either X or X^c.

□

Theorem 3.3.7: Completeness of KE

KE *is a countermodel expansion system.*

Proof

Suppose we have a **KE**-saturated set of formulas U which is not explicitly inconsistent.

First I show that there is a model satisfying each propositional formula in U. We can define a propositional model simply by:

for propositions P, **M** satisfies P if and only if $\mathbf{1}P \in U$.

A propositional model is defined by such a definition as long as no proposition must be both true and false according to the definition. By assumption, U is not explicitly inconsistent so the above definition is consistent.

Now I show that any model **M** satisfying each atomic formula in U satisfies each formula X in U. (An atomic formula is a signed formula $\mathbf{0}P$ or $\mathbf{1}P$ for a proposition P.)

I show this by induction on the structural complexity of X.

The least complex formulas are atomic formulas. By assumption, **M** satisfies all atomic formulas.

For the induction step, notice that any complex formula is either of type α or β.

Consider an α-type formula. U is (A)-saturated so contains both α_1 and α_2. By the induction hypothesis, we may assume that **M** satisfies both α_1 and α_2. Then by the semantics of propositional logic, **M** satisfies α.

Consider a β-type formula. U is (PB)-saturated so contains either β_1 or $\beta_1{}^c$. If $\beta_1 \in U$ then by the induction hypothesis, **M** satisfies β_1 and hence β. If $\beta_1{}^c \in U$ then since U is (B2)-saturated then $\beta_2 \in U$. Then by the induction hypothesis, **M** satisfies β_2 and hence β. This completes the structural induction step, and hence the proof.

□

For proof search, the rule (PB) is extremely unpleasant. It sanctions the introduction of any formula at all at any point. It may seem, therefore, that **KE** must be completely unsuitable for proof search. In fact this is not true. We can show this by introducing the following branching rules.

(PBβ_1)	$\dfrac{\beta}{\beta_1 \mid \beta_1{}^c}$	(PBβ_2)	$\dfrac{\beta}{\beta_2 \mid \beta_2{}^c}$

This rule explicitly limits which formulas are allowed to introduce new branches. It is obviously sound because it is a special case of (PB).

Looking back at the completeness proof of **KE**, you can see that the induction step for β-formulas only requires the use of the rules (PBβ_1) and (B2). Exactly the same proof, and a symmetric argument swapping 1 and 2, gives the following theorem.

Theorem 3.3.8: Completeness of analytic KE

The expansion system containing the rules (A), (B2) and (PBβ_1) is a countermodel expansion system. So is the expansion system containing (A), (B1) and (PBβ_2).

□

A suitable name for either of these expansion systems is "analytic **KE**". The name "analytic" indicates that each expansion rule can only introduce a subformula (or the complement of a subformula) of a formula that is already on the branch.

Theorem 3.3.9: Soundness of KE extended for equivalence

All the **KE** *expansion rules for equivalence are sound.*

Proof

This is straightforward, and I will not give proofs for all eight rules. As examples I will pick two rules, chosen by coin tosses, and prove them sound.

If a propositional model satisfies $\mathbf{1}X \equiv Y$ and it satisfies $\mathbf{0}Y$ then it must certainly satisfy $\mathbf{0}X$.

If a propositional model satisfies $\mathbf{0}X \equiv Y$ and it satisfies $\mathbf{1}Y$ then it must certainly satisfy $\mathbf{0}X$.

□

Theorem 3.3.10: Completeness of KE extended for equivalence

KE *extended for equivalence is a propositional countermodel system.*

Proof

The earlier completeness proof for **KE** can be used with the difference that equivalences are neither α or β-type formulas. Adding equivalences leads to another case in the induction step. This is as follows.

Consider an equivalence type formula $\mathbf{1}X \equiv Y$. U is (PB)-saturated, so contains either $\mathbf{1}X$ or $\mathbf{0}X$. If $\mathbf{1}X \in U$ then since U is (**1**E+L)-saturated, $\mathbf{1}Y \in U$. By the induction hypothesis, **M** satisfies X and Y, and hence it satisfies $X \equiv Y$. Alternatively, if $\mathbf{0}X \in U$ then since U is (**1**E-L)-saturated, $\mathbf{0}Y \in U$. By the induction hypothesis, **M** does not satisfy X or Y, and hence it does satisfy $X \equiv Y$.

The induction step for a formula $\mathbf{0}X \equiv Y$ is similar.

□

For formulas without equivalences, I observed that the completeness proof showed that only the rules (B2) and (PBβ_1) were necessary. With equivalences, we must allow (PB) to act on X in either $\mathbf{1}X \equiv Y$ or $\mathbf{0}X \equiv Y$. However, the completeness proof above shows that only this extension of analytic (PB) and the left handed equivalence rules are necessary for equivalence.

Section 3.4: Finite Complete Trees

Of course we cannot prove completeness for an arbitrary logic and expansion system. In the last section I gave a partial result which can be of help in proving completeness. In this section I show that if there is a closed **S**-tree for a set of formulas, then there is a *finite* closed **S**-tree. The proof is the standard one, using König's Lemma. The result will be very useful to me. Later in this thesis I will show the completeness of new expansion systems by transforming closed trees in an earlier system. The fact that a closed tree may be assumed to be finite will be of great help in these transformations.

First, I quote König's Lemma. A proof can be found in most books on first order logic. Fitting (1990b), for instance, proves it as Theorem 2.7.2.

Definition 3.4.1 (finitely branching tree)

A tree is *finitely branching* if every node has only a finite number of children, including the possibility that a node has no children.

Theorem 3.4.2: König's Lemma

If a finitely branching tree contains an infinite number of nodes, then it has an infinite branch.

□

Theorem 3.4.3: Finite Closed Tree Theorem

For any expansion system **S** *and set of formulas S, if there is a closed* **S**-*tree for S, then there is a finite closed* **S**-*tree for S.*

Proof

First note that no expansion system allows infinite branching. So any **S**-tree is finitely branching.

Now consider a closed **S**-tree for *S*. Every branch is closed, and must become so at a finite point. So we could prune every branch at the point at which it first becomes closed. In the resulting tree, every branch would be finite. Then, by König's Lemma, the whole tree must be finite.

□

Section 3.5: Positions in Expansion Trees

In the later parts of this thesis, when I wish to discuss transformations of one tree into another, I need to know more about formulas than just the formula itself. For example, I will need to distinguish between formulas introduced by propositional rules, those introduced by rules for universal quantifiers, and those introduced by rules for existential quantifiers. In general, once we replace the use of formulas by the use of positions, we can tell from the occurrence of a position in a tree the expansions used to introduce that position and its parents.

Notation

The name ''position'' comes from Wallen (1989). There, the name arises because formulas are replaced by positions in a formula tree.

There is another reason for the use of positions, although one that I will not exploit in this thesis. The argument applies if an expansion system contains branching rules. Suppose, for example, that it is necessary to apply two independent branching expansions. Then after the first is applied, the second must be applied once for each new branch. This, as Wallen (1989) argued, leads to considerable ''notational redundancy''. That is, the same details have to be repeated many times in a tree. By replacing the use of formulas by positions, we can avoid this need. At least notationally, we only have to perform each expansion once. Whenever the expansion is needed, we can refer, perhaps using a pointer, to the single place where the actual expansion is carried out. This idea was used by Boyer and Moore (1972), who called it ''structure-sharing''.

Formal Details of Positions

First, I assume that we have some way of referring to different *occurrences* of formulas in a tree. Such a way would be to write out the tree on paper and refer to an occurrence's location on the page. For a particular expansion tree $\mathcal{T}$, I will call the set of all *occurrences* of formulas in $\mathcal{T}$, "**Occ**".

I can now define an equivalence relation $\sim_{\mathcal{T}}$ on **Occ**, where $p \sim_{\mathcal{T}} q$ is to be read as ''p is the same position as q''. I also define a binary relation $\ll$, where $p \ll q$ is read as ''q is a subposition of p''.

Definition 3.5.1 ($\sim_{\mathcal{T}}$)

Given $\mathcal{T}$, $\sim_{\mathcal{T}}$ is the smallest binary relation on **Occ** which satisfies:

1 $\sim_{\mathcal{T}}$ is an equivalence relation.

2 Suppose the same expansion rule is applied to $(X_1^1, \ldots, X_n^1)$ yielding $(Y_1^1, \ldots, Y_m^1)$ and to $(X_1^2, \ldots, X_n^2)$ yielding $(Y_1^2, \ldots, Y_m^2)$. If each Y_i^1 and Y_i^2 are occurrences of the same formula and if each $X_i^1 \sim_{\mathcal{T}} X_i^2$, then, for each j, $Y_j^1 \sim_{\mathcal{T}} Y_j^2$.

I can now talk about equivalence classes of **Occ** under $\sim_{\mathcal{T}}$.

Definition 3.5.2 (Position, Subposition, $\ll$)

Given $\mathcal{T}$, a *position* in $\mathcal{T}$ is an equivalence class of **Occ** under $\sim_{\mathcal{T}}$. The set of all equivalence classes (i.e. the set of all positions) is called $\mathbf{P}_{\mathcal{T}}$ (or simply **P** if $\mathcal{T}$ is understood).

Notationally, I will refer to elements of **P** *as* p, p_0, p_1, p_2, *etc.*

If positions $(X_1, \ldots, X_n)$ are expanded to yield $(Y_1, \ldots, Y_m)$ then for each i and j, $X_i \ll Y_j$. We say that Y_j is a *subposition* of X_i.

For the rest of this thesis, I shall often refer to *positions* instead of *formulas*. This should cause no confusion, since positions have been defined so that each occurrence of each position is associated with precisely one polarised formula.

Following Wallen (1989), I can define two functions

$lab : \mathbf{P} \rightarrow Formulas$

$pol : \mathbf{P} \rightarrow \{ \mathbf{0}, \mathbf{1} \}$

where $lab(p)$ gives the *label* (the associated formula) of a position p and $pol(p)$ gives its *polarity*.

It is useful to regard positions as having the same type as the polarised formula they represent. So for instance if, for some position p, $lab(p) = P \wedge Q$ and $pol(p) = \mathbf{0}$, then p would be of type β.

Notation

Often the principal fact that I will be interested in about a position is its type. Where this is so, I will refer to the position by its type, possibly with a disambiguating superscript. So I might refer to an α *position as* α, *and a* β *position as* β.

On other occasions, I will be interested not in the type of a position, but in the type of the expansion that introduces a position. I will call this the ***secondary type*** *of the position. Here again, I will use this to refer to the position. For example, in the case of the expansion system* **Tab**, *I might refer to a position as* α_1, α_2, β_1, β_2.

Wherever I wish to consider the subset of positions in **P** *of a certain type, I will name this set by "***P***" with the appropriate superscript. For instance, I might refer to* $\mathbf{P}^{\alpha_1}$ *as the set of positions of secondary type* α_1.

Chapter 4

Expansion Systems for Restricted Quantification

In the earlier chapters I have introduced syntax and semantics for restricted quantification and a general way of expressing tree based proof systems. In this chapter I show how a simple extension of propositional expansion systems can be made which is always sound. However, it is not always complete, so I give a sufficient condition for its completeness. This sufficient condition, that every set of restrictions in a restriction theory must have a least Herbrand model,[1] has arisen in the work on resolution by Bürckert (1990a) and the work on substitutional reasoning by Frisch (1989).

The extension of propositional proof systems I introduce in this chapter has two essential features. First, the expansion rules for quantifiers are *unary*; that is they never split a branch into two. This means that they do not change the basic structure of proofs. Second, in the extended expansion systems, the restriction theory is only referred to by the side condition on one of the quantifier rules. This means that there is a clear syntactic separation of the restriction theory from the rest of a logic, a separation that computer implementations can take advantage of.

Completeness of the extended systems I introduce in this chapter only holds for countermodel propositional expansion systems. This is rather interesting. Although countermodel propositional expansion systems are defined purely propositionally, it turns out that this is an important property for restricted quantification. Essentially, the countermodel property allows us to delve sufficiently into the structure of formulas to use the quantifier rules whenever we need to. In non-countermodel expansion systems we may not be able to do this. Later in this thesis I show that further conditions are required of a restriction theory before we can extend a non-countermodel propositional system to restricted quantification.

In fact in this chapter I introduce a number of expansion systems, corresponding to different combinations of conditions attached to the quantifier rules. Fortunately, I need only prove soundness and completeness once: soundness for the freest system and completeness for the most controlled.

In Section 4.1 I introduce the new expansion rules for restricted quantification. Because of side conditions attached to these rules, they are in fact not expressible as expansion systems as defined in Chapter 2. I discuss this issue in Section 4.1.1. In Section 4.2 I give examples of using the extended expansion system. In Section 4.3 I show that the new rules are sound, and in Section 4.4 that they are complete. In Section 4.5, as a diversion, I show that the same proof system is correct for more restriction theories if the definition of branch closure is extended.

Section 4.1: Expansion Rules for Restricted Quantification

I will now introduce the expansion rules that will be the central object of study in this thesis.

These rules will have associated with them various *side conditions*. A side condition associated with an expansion rule means that wherever the rule is applied in a tree, the side condition must be satisfied at that point. I will also have assumptions that must be checked with reference to a whole tree, for example that all δ-expansions of the same position introduce the same name. I will call these *global conditions*.

My use of side conditions is a little inconvenient with respect to my earlier discussion of expansion systems. Fortunately, the difficulty is slight and I shall mention how to get over it when I need to.

From now on I consider the following expansion rules:

[1] I formally define least Herbrand models in Section 4.4.1.

Quantifier Expansion Rules

(γ)	$\dfrac{\gamma}{\gamma_0(t)}$	(δ)	$\dfrac{\delta}{\begin{array}{c}\delta_0(a)\\ \delta_r(a)\end{array}}$

Notation

If **S** *is any propositional expansion system, I will use "*Σ**S***" to refer to the expansion system containing all the expansion rules of* **S***, as well as* (γ) *and* (δ)*.*

Recall the definition of *positions* that I introduced in Section 3.5. The new expansion rules mean that new types of positions and subpositions may arise.

Notation

Following the notation of Section 3.5, where I am interested in the type of a position, I might refer to it as γ *or* δ*. If I am interested in the secondary type of a position, I will refer to such positions as* $\gamma_0(t)$*,* $\delta_0(a)$*, or* $\delta_r(a)$*. If the name a or term t is unimportant then I will omit it. Although not strictly defined as in Section 3.5, I will also regard* $\gamma_r(t)$ *as a subposition of a position* γ*.*

As indicated in Section 3.5 I will use, for example, $\mathbf{P}^{\gamma}$ *to refer to the set of positions of type* γ *in* **P***, and* $\mathbf{P}^{\gamma_r}$ *to refer to the set of positions of secondary type* γ_r *in* **P***.*

The side conditions I wish to consider for the moment are listed below.

Definition 4.1.1 (Side Conditions *(1) (2) (3) (4)***)**

(*1*) applies to δ-expansions:

a is a new name that has not appeared on the branch up to the point of the δ-expansion.

(*2*) applies to γ-expansions:

$\mathbf{R} \vDash_{\Sigma} \gamma_r(t)$ where **R** is the set of Σ-atoms on the branch up to the point of the γ-expansion.

(*3*) applies to γ-expansions:

t contains only names that have been introduced by δ-expansions on the branch up to the γ-expansion, except possibly the name ε.

(*4*) There is a well defined and one to one function $f{:}\mathbf{P}^{\delta}\rightarrow Names$ such that $f(\delta)$ is the name introduced into any immediate subposition of δ. That is, for each δ-type position $\delta \in \mathbf{P}$, if $\delta_0(a) \in \mathbf{P}$ then $f(\delta) = a$.

Note that (*4*) is a global condition that must be checked over whole tree simultaneously.

Notation

To indicate which side conditions apply, I will decorate the name of the system Σ**S** *with the names for the side conditions. For example, "*Σ**S**(*124*)*" refers to the expansion system whose expansion rules are those of* **S** *together with* (γ) *and* (δ)*, and in which the side condition (1) applies to* δ*-expansions, (2) applies to* γ*-expansions, and (4) applies globally.*

This chapter and the next four chapters are devoted to studying the proof system I have just introduced, or proof systems derived from it. There is an important reason for this. The underlying theory Σ is *only* used in the side condition (*2*). As a result, there is a complete separation between reasoning within and outwith the theory Σ.

Given a propositional expansion system **S**, I wish to show the soundness and completeness of the corresponding expansion system Σ**S** with some set of side conditions applying. Completeness will only be possible when **S** is a countermodel expansion system. I turn to non countermodel expansion systems

later.

I wish to prove correct several different sets of side conditions. To avoid giving several repetitive proofs, I will proceed as follows. I will show the soundness of the system $\Sigma\mathbf{S}(12)$. For all the other systems I consider, a tree in that system will also be a $\Sigma\mathbf{S}(12)$ tree, and so the soundness of $\Sigma\mathbf{S}(12)$ applies. Then I will show the completeness of $\Sigma\mathbf{S}(1234)$. Since a $\Sigma\mathbf{S}(1234)$ tree will be a tree in all the other systems I consider, completeness of all these systems will follow.

Section 4.1.1: Side Conditions Expressed as Expansion Rules

Although I express the conditions as side conditions, two of them can be expressed as expansion rules quite simply, namely (*2*) and (*3*). However, in each case, the rule would have to be expressed by enumerating (somehow!) a very large collection of possible schemas. Because this would be so tedious it is best not to do it at all. The main point is that it could be done in principle, so the expansion rule (γ) with (*2*) and/or (*3*) built in may be regarded as an ordinary expansion rule.

The expansion rule with (*2*) built in could be represented by a large collection of schemas:

$$\begin{array}{c} \delta_r^1(a_1) \\ \delta_r^2(a_2) \\ \cdot \\ \cdot \\ \delta_r^n(a_n) \\ \gamma \\ \hline \gamma_0(t) \end{array}$$

There would have to be one such schema for each possible way of ensuring that:

$$\delta_r^1(a_1), \delta_r^2(a_2), \ldots \delta_r^n(a_n) \vDash_\Sigma \gamma_r(t)$$

Similarly for the condition (*3*) we could have a large number of schemas:

$$\begin{array}{c} \delta_r^1(a_1) \\ \delta_r^2(a_2) \\ \cdot \\ \cdot \\ \delta_r^n(a_n) \\ \gamma \\ \hline \gamma_0(t) \end{array}$$

There would be one schema for every case where each name in t is in some a_i or is ε.

The two side conditions could be combined.

Unfortunately the side condition (*1*) cannot be expressed as an expansion rule. It relies on negative information, that a certain name has not appeared yet. Expansion rules cannot express this because there is no mechanism for ensuring that all formulas on a branch are used in an expansion. Similarly, it is not possible to express (*4*) as an expansion rule. For this reason, before giving the completeness theorem I will discuss how the new conditions on trees affects the Completed Tree Existence Theorem, which will in fact still hold.

For the sake of simplicity, I will still refer to systems with these side conditions as expansion systems. The proofs I give will take account of the slight problems raised above where necessary.

Section 4.2: Examples

In the following table, I consider the modal logic **K** and the modal logics in which exactly one semantic condition holds. For each condition, I give the characteristic axiom for that condition.

Some Modal Logics		
Logic	Semantic Condition	Characteristic Axiom
K	none	$\Box(A \supset B) \supset (\Box A \supset \Box B)$
D	seriality	$\Box A \supset \Diamond A$
T	reflexivity	$\Box A \supset A$
KB	symmetry	$A \supset \Box\Diamond A$
K4	transitivity	$\Box A \supset \Box\Box A$

To use Σ**S** for modal logics, it seems that we must first translate the modal formula to a Σ-logic. This removes completely all modal syntax, leading to aesthetically unpleasing proofs. Fortunately, a simple fact saves us from needing to do this. The translation from modal logic into Σ-logic acts on the structure of the formula in exactly the same way as the expansion rules do. This means that we need only apply the translation one step at a time, simultaneously with doing a single expansion. Indeed, when we get down to atomic formulas, the translation is trivial and we do not even need to apply it! However, we need to extend the syntax of derivations slightly. The translation function always needs a world argument, and we must supply this. To do so, I will introduce a binary relation "$\Vdash$" between worlds and polarised formulas. If w is a world and X a polarised formula then $w \Vdash X$ is read as "*w forces X*" or as "*X* is true at world *w*". At the start of the proof, we need to supply the translation function with a starting point. I will use "ε".

The argument from the previous paragraph could be applied to derive an expansion rule for each modal connective. This would result in a special purpose theorem proving method for modal logics.

I present proofs in Σ**Tab**(*12*) of the characteristic axioms in the appropriate theories.

Characteristic axiom of **K** *with the empty theory.*

$$\begin{array}{c}
\underline{\varepsilon \Vdash \mathbf{0}\,\Box(A \supset B) \supset (\Box A \supset \Box B)} \\
\varepsilon \Vdash \mathbf{1}\,\Box(A \supset B) \\
\varepsilon \Vdash \mathbf{0}\,\Box A \supset \Box B \\
\hline
\varepsilon \Vdash \mathbf{1}\,\Box A \\
\varepsilon \Vdash \mathbf{0}\,\Box B \\
\hline
1 \Vdash \mathbf{0}\,B \\
\varepsilon \mathcal{R} 1 \\
\hline
\dfrac{1 \Vdash \mathbf{1}\,A \supset B}{1 \Vdash \mathbf{1}\,A}\ (i) \\
\hline
\dfrac{1 \Vdash \mathbf{0}\,A}{\otimes} \qquad \dfrac{1 \Vdash \mathbf{1}\,B}{\otimes}
\end{array}$$

(*i*) (*2*) is satisfied since $\varepsilon \mathcal{R} 1$ appears on the branch.

Characteristic axiom of **D** *with* Σ = *seriality.*

$$\begin{array}{c}
\underline{\varepsilon \Vdash \mathbf{0}\,(\Box A \supset \Diamond A)} \\
\varepsilon \Vdash \mathbf{1}\,\Box A \\
\varepsilon \Vdash \mathbf{0}\,\Diamond A \\
\hline
\dfrac{next(\varepsilon) \Vdash \mathbf{1}\,A}{\dfrac{next(\varepsilon) \Vdash \mathbf{0}\,A}{\otimes}}\ (ii)
\end{array}\ (ii)$$

(*ii*) (*2*) is satisfied since $\varepsilon \mathcal{R}\, next(\varepsilon)$ is forced by seriality.

Characteristic axiom of **T** *with* Σ = *reflexivity.*

$$\begin{array}{c} \varepsilon \Vdash \mathbf{0}(\Box A \supset A) \\ \hline \varepsilon \Vdash \mathbf{1}\Box A \\ \varepsilon \Vdash \mathbf{0}A \\ \hline \varepsilon \Vdash \mathbf{1}A \\ \hline \otimes \end{array} \quad (iii)$$

(*iii*) (*2*) is satisfied since reflexivity forces $\varepsilon \mathcal{R} \varepsilon$.

Characteristic axiom of **KB** *with* Σ = *symmetry.*

$$\begin{array}{c} \varepsilon \Vdash \mathbf{0}(A \supset \Box\Diamond A) \\ \hline \varepsilon \Vdash \mathbf{1}A \\ \varepsilon \Vdash \mathbf{0}\Box\Diamond A \\ \hline 1 \Vdash \mathbf{0}\Diamond A \\ \varepsilon \mathcal{R} 1 \\ \hline \varepsilon \Vdash \mathbf{0}A \\ \hline \otimes \end{array} \quad (iv)$$

(*iv*) (*2*) is satisfied since $\varepsilon \mathcal{R} 1$ appears on the branch and symmetry ensures that $\varepsilon \mathcal{R} 1 \vDash_\Sigma 1 \mathcal{R} \varepsilon$.

Characteristic axiom of **K4** *with* Σ = *transitivity.*

$$\begin{array}{c} \varepsilon \Vdash \mathbf{0}(\Box A \supset \Box\Box A) \\ \hline \varepsilon \Vdash \mathbf{1}\Box A \\ \varepsilon \Vdash \mathbf{0}\Box\Box A \\ \hline 1 \Vdash \mathbf{0}\Box A \\ \varepsilon \mathcal{R} 1 \\ \hline 2 \Vdash \mathbf{0}A \\ 1 \mathcal{R} 2 \\ \hline 2 \Vdash \mathbf{1}A \\ \hline \otimes \end{array} \quad (v)$$

(*v*) (*2*) is satisfied since $\varepsilon \mathcal{R} 1$ appears on the branch and transitivity ensures that $\varepsilon \mathcal{R} 1,\ 1 \mathcal{R} 2 \vDash_\Sigma 1 \mathcal{R} \varepsilon$.

It should not be assumed that all modal logics can have their characteristic axiom proved in this way. For example, the modal logic **S4.3** has an accessibility theory that cannot be written in Horn clauses and fails the least Herbrand model property. No proof for its characteristic axiom could be constructed in Σ**S**(*12*) for any sound expansion system **S**. In Chapter 9 I introduce a proof system that can be applied to such logics, and in Section 9.2 give the characteristic axiom of **S4.3** as an example.

Section 4.3: Soundness of ΣS(*12*)

The definition of soundness of an expansion system means that I need only check that the two rules (γ) and (δ) are each sound, where the side conditions (*1*) and (*2*) hold. Although (δ) with the side condition (*1*) is not, strictly speaking, an expansion rule, the distinction is unimportant in this context.

Theorem 4.3.1: Soundness of (γ)

(γ) is a sound expansion rule if the side condition (*2*) *holds.*

Proof

Suppose we have a set S containing a formula γ that may be expanded under the rule (γ) to yield $\gamma_0(t)$. Suppose also that S is satisfiable, and in particular that it is satisfiable in a Σ-model $\mathbf{M} = \langle \mathbf{D},\mathbf{I} \rangle$ under an assignment **A**. That is, $\gamma^{\mathbf{I},\mathbf{A}} = \mathbf{t}$.

By the side condition (*2*), $\mathbf{R} \vDash_\Sigma \gamma_r(t)$, where **R** is the set of restrictions in S. Since **M** is a Σ-model of S, **M** is a Σ-model of **R** and hence of $\gamma_r(t)$. That is, $\gamma_r(t)^{\mathbf{I},\mathbf{A}} = \mathbf{t}$.

We have that $\gamma^{\mathbf{I},\mathbf{A}} = \mathbf{t}$ and that $\gamma_r(t)^{\mathbf{I},\mathbf{A}} = \mathbf{t}$. By the definition of semantics of Σ-logic, and since **A** is certainly a variant of itself, $\gamma_0(t)^{\mathbf{I},\mathbf{A}} = \mathbf{t}$. Therefore **M** and **A** satisfy S, $\gamma_0(t)$, which was what was wanted.

□

Theorem 4.3.2: Soundness of (δ)

(δ) is a sound expansion rule if the side condition (1) is satisfied.

Proof

Suppose we have a set S containing a formula δ that may be expanded under the rule (δ) to yield $\delta_0(a)$. Since the expansion must satisfy the side condition (*1*), we know that the name a does not appear anywhere in S. Suppose also that S is satisfiable, and in particular that it is satisfiable in a Σ-model $\mathbf{M} = <\mathbf{D},\mathbf{I}>$ under an assignment **A**. Then $\delta^{\mathbf{I},\mathbf{A}} = \mathbf{t}$.

By the definition of semantics of Σ-logic, there is an assignment **B** such that $\delta_0^{\mathbf{I},\mathbf{B}} = \mathbf{t}$ and $\delta_r^{\mathbf{I},\mathbf{B}} = \mathbf{t}$. Suppose that the outermost variable x in δ is mapped to $d \in \mathbf{D}$ by **B**.

Now, I define a new interpretation **J**. **J** is defined to be exactly the same as **I** except that **J** maps a to d. Since a appears nowhere in S and S is satisfied by **I**, S is satisfied by **J**. Furthermore, $\delta_0(a)^{\mathbf{J},\mathbf{A}} = \delta_0(x)^{\mathbf{I},\mathbf{B}} = \mathbf{t}$ and $\delta_r(a)^{\mathbf{J},\mathbf{A}} = \delta_r(x)^{\mathbf{I},\mathbf{B}} = \mathbf{t}$. Therefore $\mathbf{M}' = <\mathbf{D},\mathbf{J}>$ and **A** satisfy S, $\delta_0(t)$, which was what was wanted.

□

Section 4.4: Completeness of ΣS(*1234*)

Recall that ΣS(*1234*) is the expansion system containing the rules in **S** as well as the rules (γ) and (δ) which must satisfy the conditions (*1*), (*2*), (*3*), and (*4*). In this section I prove the completeness of ΣS(*1234*) provided that Σ satisfies what I call the "least Herbrand model property" (defined in this section) and that **S** is a countermodel expansion system. Of course if we drop one or more of the side conditions then the resulting proof system is still complete. In particular, the completeness of ΣS(*1234*) is enough to show the completeness of ΣS(*12*), ΣS(*123*), and ΣS(*124*).

In Chapter 9 I introduce suitable proof systems if the least Herbrand model property does not hold.

As well as a condition on Σ, completeness requires that **S** be a countermodel expansion system. Completeness fails on very simple examples in non-countermodel systems.

Example (failure of* ΣS(*1234*) *in non-countermodel systems)

Resolution serves as a typical example of a non-countermodel expansion system. This example is so simple that the precise details of a resolution system are unimportant, as are the details of the restriction theory (it may be empty, for example). Note that I am only considering the *propositional* resolution rule. Consider the formula

$$\exists x_{:\rho(x)}\, \mathbf{f} \vee \exists y_{:\rho(y)}\, \mathbf{f}$$

Of course this formula is unsatisfiable, and could be shown to be so by (say) ΣKE(*1234*). However, the resolution rule cannot apply to the formula since there are no other formulas present. Also, the rule (δ) cannot apply since the dominating connective is propositional. Therefore no resolution proof can be found.

This example shows that the propositional resolution rule cannot, naively, be extended to deal with restricted quantification using the rules I have introduced. Frisch (1991) has shown how resolution can be extended for a class of restriction theories: I arrive at a similar conclusion by a different route in Chapter 8, though I do not consider resolution explicitly.

End of Example

Because of the conditions (*1*) and (*4*) the Completed Tree Existence Theorem does not apply directly, as it did in Chapter 2. So first I discuss the slight changes needed in its proof.

Theorem 4.4.1: Completed Tree Existence Theorem for ΣS(*1234*)

There is a completed Σ**S**(*1234*) *tree for any finite set of formulas S.*

Proof

The central feature of the earlier proof of this theorem was that if an expansion is possible in U then it is possible in $U \cup \{X\}$ if X is a formula introduced from U by a different expansion. This property holds trivially for expansion rules as defined earlier. As the rule (γ) with the side conditions (*2*) and (*3*) can be seen as an expansion rule, it causes no problems. Also, (*4*) causes no problems: it gives us a function f, and if $\delta \in U$ it sanctions the introduction of $\delta_0(f(\delta))$ from both U and U, X.

The only problem then is the side condition (*1*) for the rule (δ). But suppose δ may be introduced from U: that is $\delta \in U$ and the name $f(\delta)$ does not appear in U. If a propositional expansion rule applies to introduce X then it introduces no new names and the δ-expansion is still valid. If (γ) applies then, from (*3*), X contains no names not in U, and hence not $f(\delta)$. If (δ) applies then unless it applies to δ itself, it applies to some other position and by (*4*) introduces a different name. In either case the δ-expansion still is applicable to $U \cup \{X\}$.

The rest of the proof is as before.

□

Section 4.4.1: Least Herbrand Models

One essential feature of the expansion rules I have introduced is the unary nature of the (γ) rule. If it were a branching rule then the proof system would be more difficult for humans to use, and the proof-theoretical arguments I give in this thesis would be more difficult and probably weaker. In both cases this would be because extra branching structure would be introduced into expansion trees, making them much more difficult to deal with and reason about. Therefore it is crucial to know when the (γ) rule is satisfactory. It is always sound, given (*2*), but Σ**S**(*1234*) is only complete for certain theories Σ. To describe these theories, I will introduce the notion of a *least Herbrand model*.

Definition 4.4.1.1 (Herbrand model, least Herbrand model)

A Herbrand model is a model in which the domain consists of the variable free *terms*, and in which the interpretation function is the identity: the interpretation of any term is that term itself.

A Herbrand model is a *least Herbrand model* of a set **R** of Σ-atoms iff the only Σ-atoms true in the model are the logical consequences of the set **R** (that is ρ is in the least Herbrand model iff $\mathbf{R} \models_\Sigma \rho$).

Definition 4.4.1.2 (least Herbrand model properties)

Σ has the *least Herbrand model property* if any set of variable free Σ-atoms has a least Herbrand Σ-model.

Σ has the *weak least Herbrand model property* if any set of variable free Σ-atoms *either* is Σ-unsatisfiable *or* has a least Herbrand Σ-model.

Theorem 4.4.1.3: Least Herbrand Uniqueness

If a least Herbrand model exists for a set of literals, it is uniquely defined on all formulae containing only those literals.

Proof

A model is defined by its assignment of truth to literals. Since this is fixed by the definition of least Herbrand model, the model is fixed.

□

Least Herbrand models are in fact very familiar, since anything constructed by pure Prolog is a least Herbrand model. This follows from:

Theorem 4.4.1.4: Horn Clauses and the Least Herbrand Model Property

(i) *If* Σ *is written in Horn clauses, then* Σ *satisfies the weak least Herbrand model property.*

(ii) *If* Σ *is written in definite Horn clauses, then* Σ *satisfies the least Herbrand model property.*

Sketch Proof

I will sketch the argument given by Lloyd (1984, up to Theorem 6.2). First observe that Σ-atoms may be considered as clauses, and indeed as facts, and therefore adding them to Σ leaves a set of Horn clauses (in (i)) or definite Horn clauses (in (ii)).

One can identify a Herbrand model with the set of variable free atoms true in that model. Then one can show that, if Σ is in Horn clause form, the intersection of a set of Herbrand models for Σ is also a Herbrand model for Σ. But ϕ is a consequence of Σ if and only if ϕ is true in all Herbrand models of Σ. Therefore, if Σ has any Herbrand model, the intersection of all Herbrand models for Σ will be a least Herbrand model. This proves (i).

Part (ii) is a simple Corollary of (i), from the observation that the set of all variable free atoms is a Herbrand model of Σ, if Σ is written in definite Horn clauses.

□

To illustrate the concept further, examples abound were the least Herbrand model property fails. For example, suppose Σ is:

$$number(X) \supset (\ even(X) \vee odd(X)\ )$$

Then even the single atom

$$number(n)$$

does not have an least Herbrand Σ-model. If it did have a least Herbrand model then either "$even(n)$" or "$odd(n)$" would have to be a Σ-consequence of Σ and $number(n)$, which is ridiculous.

The converse of Theorem 4.4.1.4 is not true. There are non-Horn theories which satisfy the least Herbrand model property. For example, take the simple theory:

$$\rightarrow\ p(X), p(X).$$

This is not a Horn clause, having two literals after the arrow. However an equivalent theory can certainly be written in Horn clauses, namely:

$$\rightarrow\ p(X).$$

Indeed any theory satisfying the least Herbrand model property can be written in Horn clauses.[2] I will not prove this here as the point is not central to this thesis. However, it does mean that one can safely equate the concept of the least Herbrand model property with the ability to be written in definite Horn clauses.

[2] I am grateful for a conversation with Wilfred Chen that clarified this point for me.

Section 4.4.2: Completeness Theorem

There are two aspects to proving the completeness theorem. First I show how, given an open complete branch, I can define a (well-defined) Σ-model. Then I show that this model in fact satisfies every formula on the branch. I will prove these separately because I will later wish to prove completeness for other proof systems, and it will turn out that only the well-definedness of the model will need to be reproved.

Definition 4.4.2.1 ($\mathbf{M}_\lambda$, Σ-model for a branch)

Suppose that λ is a branch in a Σ**S** tree, containing the set of formulas U and set of restrictions **R**. In that case, $\mathbf{M}_\lambda = \langle \mathbf{D},\mathbf{I} \rangle$ is a Σ-*model for the branch* λ if:

1. **D** is the set of all terms built using function symbols from the constants and names that appear anywhere in U; and
2. **I** applied to constant symbols and names is the identity function; and
3. **I** applied to a n-place function symbol f gives the function defined by[3]

$$f^{\mathbf{I}}(t_1, \ldots t_n) = f(t_1, \ldots t_n)$$

4. for any γ-type formula γ on the branch, and any term t arising on the branch, $\mathbf{M}_\lambda$ satisfies the associated restriction $\gamma_r(t)$ if and only if $\mathbf{R} \vDash_\Sigma \gamma_r(t)$; and
5. for any P, an $\mathcal{L}$-atom, **M** satisfies P if and only if $\mathbf{1}P \in U$; and
6. $\mathbf{M}_\lambda$ is a Σ-model.

Part (4) of the definition means that there may be many models $\mathbf{M}_\lambda$ for a branch. However, they will only differ on restrictions that are unimportant to satisfying the set of formulas on the branch.

Lemma 4.4.2.2: Branch Model Lemma for ΣS**(*1234*)**

If Σ satisfies the least Herbrand model property, and λ is an open complete branch in a Σ**S**(*1234*) tree then there is at least one Σ-model $\mathbf{M}_\lambda$ for the branch λ.

Proof

By the least Herbrand model property, **R** has a least Herbrand Σ-model $\mathbf{M}_0$. The Σ-atoms required to be true (or false) by part (4) of Definition 4.4.2.1 are all contained (or not contained) in $\mathbf{M}_0$. We can extend $\mathbf{M}_0$ simply by defining $\mathbf{M}_\lambda$ by (5) of Definition 4.4.2.2 for $\mathcal{L}$-atoms. This does not affect any Σ-atoms, so $\mathbf{M}_\lambda$ is a Σ-model. It is well defined since any first order logic model is well defined given the definition of which atoms hold in it.

□

Theorem 4.4.2.3: Completeness of ΣS**(*1234*)**

If Σ satisfies the least Herbrand model property, and **S** *is a propositional countermodel expansion system, and a finite set of formulas S is not satisfiable, then there is a closed* Σ**S**(*1234*)*-tree for S.*

Proof

I show the contrapositive. That is, assuming that there is no closed Σ**S**(*1234*)-tree for a set of formulas S, I show that S is satisfiable. The first step is to use the Completed Tree Existence Theorem to give us a completed Σ**S**(*1234*)-tree for S. Since this tree cannot be closed, it contains an open, complete, branch. Call this branch λ, and the set of formulas on it U. Suppose the set of all restrictions that appear in U is **R**. From the Branch Model Lemma there is a Σ-model for the branch: I assume we pick a particular Σ-model $\mathbf{M}_\lambda$.

[3] Note that **I** has no visible effect when applied to a term.

Since **S** is a propositional expansion system, no expansion rule in **S** can introduce a formula with greater quantifier depth than its parents. Also, the expansion rules for quantifiers (γ- and δ-type formulas) introduce formulas of strictly smaller quantifier depth. These two facts imply that if we consider all formulas in U of less than a certain quantifier depth, that set must be $\Sigma\mathbf{S}(1234)$-saturated. This is crucial to the following induction argument.

It remains to show that for any polarised formula X in λ, $\mathbf{M}_\lambda$ satisfies X. I proceed by induction on the number of quantifiers in X.

The base follows from propositional completeness. That is, if X is quantifier free, then we consider all formulas in U containing no quantifiers. This set is **S**-saturated, open, and contains X. Since **S** is a countermodel expansion system $\mathbf{M}_\lambda$ satisfies X.

As the induction hypothesis I assume that all formulas X containing only a certain number k of quantifiers are satisfied by $\mathbf{M}_\lambda$. For the step I first consider quantifications containing $k+1$ quantifiers, and then all formulas containing $k+1$ quantifiers.

Any quantification is either a γ-type formula or a δ-type formula.

Consider a γ-type formula γ appearing on the branch, and any element of the domain - that is any term t. Now $\mathbf{M}_\lambda$ satisfies $\gamma_r(t)$ if and only if $\mathbf{R} \models_\Sigma \gamma_r(t)$. So if $\mathbf{M}_\lambda$ satisfies $\gamma_r(t)$, then $\mathbf{R} \models_\Sigma \gamma_r(t)$, and the expansion rule for γ-type formulas applies to γ, with image $\gamma_0(t)$. By definition of **D**, every name in t appears in U, and so the expansion satisfies (*3*). Since U is saturated under the expansion rule for γ-type formulas, $\gamma_0(t) \in U$. By the induction hypothesis, $\gamma_0(t)$ is satisfied by the model. This establishes that for all domain elements t such that $\gamma_r(t)$ holds, $\gamma_0(t)$ holds. Hence $\mathbf{M}_\lambda$ satisfies γ.

Consider a δ-type formula δ appearing on the branch, and the name $f(\delta)$ given to us by (*4*). Suppose that $\delta_0(f(\delta))$ and $\delta_r(f(\delta))$ do not appear in U. Then certainly no other δ-expansion on the branch can introduce the name $f(\delta)$ from (*4*). But then no γ-expansion on the branch could introduce $f(\delta)$, as each γ-expansion satisfies (*3*). But then U, $\delta_r(f(\delta))$, $\delta_0(f(\delta))$ would be a possible expansion from U and the expansion would satisfy (*4*) and (*1*). Since U is $\Sigma\mathbf{S}(1234)$ saturated, we know that $\delta_r(f(\delta)), \delta_0(f(\delta)) \in U$. By the induction hypothesis, both formulas are satisfied by $\mathbf{M}_\lambda$. Hence $\mathbf{M}_\lambda$ satisfies δ.

To complete the induction step, we look at U in a new way. We consider only the propositional rules in **S**, but as well as any atom, we regard any quantification as an atomic proposition. We can do this since no propositional rule can act on a quantification. We know from the induction base that each atom is satisfied in $\mathbf{M}_\lambda$. From the induction hypothesis and the arguments of the previous two paragraphs, we know that $\mathbf{M}_\lambda$ satisfies each quantification containing $k+1$ or fewer quantifications. Thus, regarding $\mathbf{M}_\lambda$ as a propositional model, it satisfies all the formulas we are treating as atomic. As I observed earlier, the set of formulas in U with $k+1$ quantifiers or fewer is **S**-saturated. Hence, since **S** is a countermodel expansion system, all formulas containing at most $k+1$ quantifications are satisfied by $\mathbf{M}_\lambda$.

This completes the induction step, and hence the completeness theorem.

□

Section 4.5: A Slightly More General System

The theorem I have just proved will form the foundation for all the work in Chapters 5, 6, 7, and 8. However, I wish to prove a slight generalisation of the theorem in the case that Σ satisfies only the *weak* least Herbrand model property. In Chapter 9 I will generalise still further, and the following result will be a better starting point than the previous one.

The only change I wish to make is to allow for the possibility that a branch may contain an explicit contradiction given by the restrictions on a branch. In terms of Section 4.4.1, the change is to allow goal clauses into the restriction theory Σ. Everything will be the same except that I must use a more liberal definition of closure of a branch, which I call ''Σ-closure''.

Definition 4.5.1 (Σ-closure)

A branch in a Σ**S** tree is Σ-*closed* if either it is closed or if the set of restrictions on the branch is Σ-unsatisfiable.

Changing the definition of closure technically invalidates much of the work I have done up to now. However, everything is alright because Σ-closure is a *monotonic* property: if a branch is Σ-closed then any extension of it is Σ-closed. All relevant properties of closure carry through.

The crucial point is to prove the Branch Model Lemma for the new version. Indeed, after that the proof of the completeness theorem will be identical to the earlier proof.

Lemma 4.5.2: Branch Model Lemma for ΣS**(*1234*) with Σ-closure**

If Σ satisfies the weak least Herbrand model property, and λ is an open complete branch in a Σ**S**(*1234*) tree then there is at least one Σ-model $\mathbf{M}_\lambda$ for the branch λ.

Proof

By the weak least Herbrand model property, either **R** is Σ-unsatisfiable or it has a least Herbrand Σ-model. But if **R** were Σ-unsatisfiable then λ would be Σ-closed, which it is not. Therefore **R** has a least Herbrand model $\mathbf{M}_0$. The Σ-atoms required to be true (or false) by part (4) of Definition 4.4.2.1 are all contained (or not contained) in $\mathbf{M}_0$. We can extend $\mathbf{M}_0$ simply by defining $\mathbf{M}_\lambda$ by (5) of Definition 4.4.2.2 for $\mathcal{L}$-atoms. This does not affect any Σ-atoms, so $\mathbf{M}_\lambda$ is a Σ-model. It is well defined since any first order logic model is well defined given the definition of which atoms hold in it.

□

Theorem 4.5.3: Completeness of ΣS**(*1234*) with Σ-closure**

If Σ satisfies the weak least Herbrand model property, and **S** *is a propositional countermodel expansion system, and a finite set of formulas S is not satisfiable, then there is a Σ-closed Σ***S**(*1234*)*-tree for S.*

Proof

The proof is the same as that of Theorem 4.4.2.3.

□

Example (the use of Σ-closure)

A simple clause that we may wish to introduce is that for irreflexivity, which in first order logic is expressed as:

$$\forall x \neg x \mathcal{R} x$$

and as a *goal* clause is:

$$x \mathcal{R} x \rightarrow$$

For example, we may consider the modal logic containing irreflexivity, symmetry and transitivity. It is not surprising that this logic is not familiar from the literature, since the formula □**f** is valid! This is shown by the following tree which uses Σ-closure:

$$\begin{array}{c} \varepsilon \Vdash \mathbf{0}\Box\mathbf{f} \\ \hline \varepsilon \mathcal{R} a \\ a \Vdash \mathbf{0}\mathbf{f} \\ \hline \otimes \end{array}$$

The branch is Σ-closed because the single restriction $\varepsilon \mathcal{R} a$ is unsatisfiable in the chosen theory. From symmetry we can deduce from it $a \mathcal{R} \varepsilon$. From transitivity we can then deduce $\varepsilon \mathcal{R} \varepsilon$, contradicting irreflexivity.

End of Example

Chapter 5

Alphabetical Monotonicity and Non-Vacuity

In Chapter 4 I proved that for any restriction theory satisfying the least Herbrand model property, and any propositional countermodel expansion system **S**, the proof system **ΣS**(*12*) is sound and complete; and indeed it remains so if any combination of the conditions (*3*) and (*4*) are added. This result is very useful because of the simplicity of the proof system and the generality and simplicity of the conditions that make it true.

For automated proof search, three of the side conditions from Chapter 4 cause potential problems. The side condition (*2*) might require the use of the exact set of restrictions on a branch. Therefore it might be very important where exactly on a branch a γ-expansion is applied. Similarly the side condition (*3*) uses the fact that a particular set of names is on a branch, again being highly dependent on where the expansion is applied. The condition (*1*) requires a certain name not to be on a branch, again dependent on the placement of the application. The last of these problems is familiar from classical unrestricted first order logic. I discuss it further later in this thesis. In this chapter I address the first two of these problems.

The feature of this chapter is the introduction of a new side condition (*5*) to replace (*2*) for γ-expansions. Instead of using the set of restrictions on a single branch, (*5*) uses the set of restrictions on an entire tree. In fact the implications go further. Where using (*5*) is sound, we may use any set of restrictions that is a superset of the restrictions on the branch and a subset of the restrictions on the whole tree. In other words, except that we must use at least all the formulas on the branch, we have a free choice of the set of restrictions we may use. The order dependence arising from (*2*) has been eliminated. In Chapter 7 I show how this can be used to derive a matrix characterisation of validity.

In this chapter I give two sufficient conditions for the soundness of (*5*). The first condition, called ''alphabetical monotonicity'' is sufficient if (*3*) is also used. However, (*3*) also introduces some order dependence, raising problems discussed further in Chapter 7. So I also give a second condition, called ''non-vacuity'', which is sufficient whether or not (*3*) is used. Non-vacuity turns out to be more interesting than alphabetical monotonicity. I show in Chapter 6 that, where both hold, non-vacuity is easier to prove directly than alphabetical monotonicity. Also, in Chapter 8 the result on non-vacuity from this chapter serves as a basis for deriving proof systems for restricted quantification for any expansion system, not just countermodel ones. Also non-vacuity, in a slightly different form, has turned up before in the literature: it is essentially the same as Frisch's (1991) use of ''Σ-satisfiability''.

The greater order independence achieved when non-vacuity holds is shown to be of importance in Chapters 7 and 8. The basic reason for this is that with non-vacuity, reasoning about quantifiers can be completely separated out from propositional reasoning. Put simply, instantiation commutes with propositional logic, as I show in Chapter 8. This observation is not at all original with me, but is implicit in Frisch (1991) and indeed dates back to Hailperin who writes (1957b, page 114):

> ''... one readily surmises that the difficulties stem from the admission of vacuous ranges for the variables. Accordingly, then, by requiring our variables to have non-vacuous ranges we should be able to restore the desired manipulative freedom - such a limitation to non-vacuous ranges is, moreover, quite natural since in most significant deductive systems the variables have non-empty ranges.''

If we could indeed require variables to have non-vacuous ranges then I would not need to discuss alphabetical monotonicity at all. However, examples such as non-serial modal logics show that sometimes it is necessary.

Section 5.1: The Expansion System $\Sigma\mathbf{S}(1345)$

In this section I introduce a new proof system, called $\Sigma\mathbf{S}(1345)$, which is similar to $\Sigma\mathbf{S}(1234)$ but with a different side condition on γ-expansions. The main usefulness of $\Sigma\mathbf{S}(1345)$ is its role as the basis of the matrix characterisation of validity I develop in Chapter 7. However, presenting it now clearly shows its interesting features. Furthermore, it separates the theoretical development into stages making that development, I trust, easier to understand.

The only difference between $\Sigma\mathbf{S}(1345)$ and $\Sigma\mathbf{S}(1234)$ is that a new side condition (*5*) replaces (*2*). To define the new side condition I need a preliminary definition.

Definition 5.1.1 ($LHM_\Sigma(\mathcal{T})$)

If Σ satisfies the least Herbrand model property and $\mathcal{T}$ is a $\Sigma\mathbf{S}$ expansion tree, define

$$LHM_\Sigma(\mathcal{T})$$

as the least Herbrand Σ-model of the set $\mathbf{P}^{\delta_r}$. That is, it is the least Herbrand model of all the Σ-literals introduced by δ-expansions anywhere in $\mathcal{T}$. Such a model exists by the least Herbrand model property, and is unique by the Least Herbrand Uniqueness Theorem. Recall that the definition of least Herbrand Σ-model means that $\rho \in LHM_\Sigma(\mathcal{T})$ if and only if $\mathbf{P}^{\delta_r} \vDash_\Sigma \rho$.

I now define the system $\Sigma\mathbf{S}(1345)$. The conditions (*1*), (*3*), and (*4*) are as defined in Chapter 4.

Quantifier Expansion Rules for $\Sigma\mathbf{S}(1345)$

(γ)	$\dfrac{\gamma}{\gamma_0(t)}$	(δ)	$\dfrac{\delta}{\delta_0(a) \quad \delta_r(a)}$

Definition 5.1.2 (Side Condition (*5*)**)**

(*5*) applies to γ-expansions:

provided $\gamma_r(t) \in LHM_\Sigma(\mathcal{T})$

The condition (*5*) is curious: it is a *local* check that requires *global* information to be confirmed. This chapter is dedicated to the consideration of the theoretical problems this raises.

The proof systems using (*5*) instead of (*2*) do not seem to of great significance in themselves. However, they would be preferable to implement than the system of Chapter 4, since any proof tree in them is a proof tree in the earlier system, but not vice versa. There is a nice way of seeing the difference if you are familiar with Prolog. One might choose to implement the expansion rule (δ), which introduces onto a branch an explicit restriction, by asserting (in the Prolog sense) the restriction, and using Prolog to do reasoning in the restriction theory. If one did this, however, on moving to a different branch one would have to retract (in the Prolog sense) any restrictions no longer on the branch. This is an expensive operation. However, in the proof systems of this chapter one need not be careful about where in a proof tree restrictions come from: indeed it is sound to use all restrictions. That is, when moving from one branch to another one would not have to retract anything.

Earlier, it was much easier to prove the soundness of $\Sigma\mathbf{S}(1234)$ than the completeness. By contrast, the problem now is the soundness of $\Sigma\mathbf{S}(1345)$. Because it is easier to satisfy the condition (*5*) than (*2*), completeness is easy given the proven completeness of $\Sigma\mathbf{S}(1234)$. I will deal with this first.

Theorem 5.1.3: Completeness of $\Sigma\mathbf{S}(1345)$

If ϕ *is a* Σ*-formula for a theory* Σ *that satisfies the least Herbrand model property, then:*

if ϕ *is* Σ*-valid then there is a closed* $\Sigma\mathbf{S}(1345)$ *tree for* ϕ.

Proof

By the Completeness Theorem of $\Sigma\mathbf{S}(1234)$, there is a closed $\Sigma\mathbf{S}(1234)$ tree $\mathcal{T}$ for ϕ. But by definition of the side conditions (2) and (5), if a γ-expansion satisfies (2) then it certainly satisfies (5). So $\mathcal{T}$ is also a $\Sigma\mathbf{S}(1345)$ tree.

□

Completeness is trivial, but soundness is not. Indeed, for some theories Σ soundness can fail. An example is the modal theory consisting of symmetry and transitivity.

Example (Failure of soundness of $\Sigma\mathbf{S}(1345)$***)***

Suppose Σ contains only the clauses for symmetry and transitivity. Consider the modal logic formula

$$\Box\,\mathbf{t} \wedge \Diamond\,\mathbf{t}$$

which translates into Σ-logic as:

$$(\forall x_{:\varepsilon\mathcal{R}x}\,\mathbf{t}) \wedge (\exists y_{:\varepsilon\mathcal{R}y}\,\mathbf{t})$$

In an attempt to show its validity, We can look for a closed $\Sigma\mathbf{Tab}(1234)$ tree for this formula, but we cannot find one. If we try, we come up with:

$$\begin{array}{c}
\mathbf{0}\,(\forall x_{:\varepsilon\mathcal{R}x}\,\mathbf{t}) \wedge (\exists y_{:\varepsilon\mathcal{R}y}\,\mathbf{t}) \\ \hline
\begin{array}{c}
\mathbf{0}\,\forall x_{:\varepsilon\mathcal{R}x}\,\mathbf{t} \\ \hline
\mathbf{0}\,\mathbf{t} \\ \hline
\varepsilon\mathcal{R}a \\ \hline
\otimes
\end{array}
\qquad
\mathbf{0}\,\exists y_{:\varepsilon\mathcal{R}y}\,\mathbf{t}
\end{array}$$

This tree is not expandable, because on the right hand branch, for no term t (including the name ε) does $\varnothing \vDash_{\Sigma} \varepsilon\mathcal{R}t$, so no possible γ-expansion can satisfy the condition (2) in $\Sigma\mathbf{S}(12)$. So the tree is not closed. Indeed this is the only tree there is for the original formula, so the original formula is not valid.

We can find a closed tree if we allow ourselves to introduce positions that are in $LHM_{\Sigma}(\mathcal{T})$. In particular, note that the above tree contains the Σ-literal $\varepsilon\mathcal{R}a$. The least Herbrand model of $\{\,\varepsilon\mathcal{R}a\,\}$ contains $\varepsilon\mathcal{R}\varepsilon$ as the following proof shows:

$$\varepsilon\mathcal{R}a \rightarrow a\mathcal{R}\varepsilon \qquad \text{(Symmetry)}$$

$$\varepsilon\mathcal{R}a\,,\,a\mathcal{R}\varepsilon \rightarrow \varepsilon\mathcal{R}\varepsilon \qquad \text{(Transitivity)}$$

Given this, we can extend the above tree to produce a closed $\Sigma\mathbf{S}(1345)$ tree for the original formula

$$\begin{array}{c}
\mathbf{0}\,(\forall x_{:\varepsilon\mathcal{R}x}\,\mathbf{t}) \wedge (\exists y_{:\varepsilon\mathcal{R}y}\,\mathbf{t}) \\ \hline
\begin{array}{c}
\mathbf{0}\,\forall x_{:\varepsilon\mathcal{R}x}\,\mathbf{t} \\ \hline
\mathbf{0}\,\mathbf{t} \\ \hline
\varepsilon\mathcal{R}a \\ \hline
\otimes
\end{array}
\qquad
\begin{array}{c}
\mathbf{0}\,\exists y_{:\varepsilon\mathcal{R}y}\,\mathbf{t} \\ \hline
\mathbf{0t} \\ \hline
\otimes
\end{array}
\end{array}$$

The introduction of **0t** on the right hand branch satisfies (5) because $\varepsilon\mathcal{R}\varepsilon$ is in $LHM_{\Sigma}(\mathcal{T})$.

End of Example

So $\Sigma\mathbf{Tab}(1345)$ is unsound and in general $\Sigma\mathbf{S}(1345)$ can be unsound. The next question is, can we characterise when $\Sigma\mathbf{S}(1345)$ is sound and when it is not? I do not know of a *simple* characterisation of the soundness of $\Sigma\mathbf{S}(1345)$. Instead, I will proceed as follows. First, I will provide a characterisation of the

soundness of $\Sigma\mathbf{S}(1345)$. This will be essentially trivial since it just demands that the conditions (5) and (2) are equivalent for a given Σ. This is not very helpful. However, it does provide us with a benchmark. Then I will show that there are two much simpler *sufficient* conditions that guarantee the soundness of $\Sigma\mathbf{S}(1345)$ for a particular theory. In Chapter 6, I shall show that these conditions are met for a wide range of restriction theories.

Definition 5.1.4 (Negative Monotonicity Condition)

Given that Σ satisfies the least Herbrand model property, and that ϕ is a restricted Σ-logic formula, Σ satisfies the *Negative Monotonicity Condition at* ϕ if and only if,

for any $\Sigma\mathbf{S}(1345)$ tree $\mathcal{T}$ for ϕ, and for any expansion of γ to $\gamma_0(t)$ in $\mathcal{T}$, then

$$\mathbf{R} \nvDash_\Sigma \gamma_r(t) \Rightarrow \gamma_r(t) \notin LHM_\Sigma(\mathcal{T}),$$

where $\mathbf{R}$ is the set of Σ-formulas on the branch up to the γ-expansion.

Notation

I chose the name "Negative Monotonicity" for this condition because it demands that, once γ is available to be reduced, the negative fact that $\gamma_r(t)$ is not satisfied is not affected by anything else that can happen in the tree.

The example above gives an example of a theory that fails to satisfy negative monotonicity. On the right hand branch of the tree, we have $\mathbf{R}$ as the empty set $\varnothing$, and $\varnothing \nvDash_\Sigma \varepsilon \mathcal{R} \varepsilon$. However, because $\varepsilon \mathcal{R} a$ appears on the right hand branch, we have that $\varepsilon \mathcal{R} \varepsilon \in LHM_\Sigma(\mathcal{T})$. In this case, in the above definition we must set

$$\phi = \mathbf{0}\,(\forall x_{:\varepsilon \mathcal{R} x}\, \mathbf{t}) \wedge (\exists y_{:\varepsilon \mathcal{R} y}\, \mathbf{t}),$$

$$\gamma = \mathbf{0}\,\exists y_{:\varepsilon \mathcal{R} y}\, \mathbf{t}, \textit{ and}$$

$$t = \varepsilon.$$

Later in this chapter I introduce conditions on theories, namely "alphabetical monotonicity" and "non-vacuity" that both imply that negative monotonicity holds. In Chapter 6 I show that many theories satisfy these conditions, and hence negative monotonicity.

The Negative Monotonicity Condition justifies the soundness of $\Sigma\mathbf{S}(1345)$ rather trivially.

Theorem 5.1.5: Soundness of $\Sigma\mathbf{S}(1345)$

If ϕ is a restricted Σ-formula for a theory Σ that satisfies the least Herbrand model property, and Σ satisfies the Negative Monotonicity Condition at ϕ, then

if there is a closed $\Sigma\mathbf{S}(1345)$ tree for ϕ, then ϕ is Σ-valid.

Proof

Suppose $\mathcal{T}$ is a closed $\Sigma\mathbf{S}(1345)$ tree for ϕ. Consider any γ-expansion of γ to $\gamma_0(t)$. We know, from the condition (5) that $\gamma_r(t) \in LHM_\Sigma(\mathcal{T})$. Then, by the contrapositive of the Negative Monotonicity Condition, $\mathbf{R} \vDash_\Sigma \gamma_r(t)$, where $\mathbf{R}$ is the set of Σ-formulas on the branch up to the γ-expansion. So the expansion satisfies the condition (2). Hence $\mathcal{T}$ is a $\Sigma\mathbf{S}(1234)$ tree, and the result follows from the soundness of $\Sigma\mathbf{S}(1234)$.

□

The Negative Monotonicity Condition is rather unpleasant because it involves consideration of all trees for a particular formula. Fortunately, there are much simpler sufficient conditions which ensure that the Negative Monotonicity Condition is satisfied for a wide range of formulas of theories. In particular, these sufficient conditions will be enough for all formulas of order sorted logics (at least for order sorted logics as I have defined them in Section 2.6), and for all formulas of a wide range of modal logics.

Section 5.2: Alphabetical Monotonicity and the Soundness of $\Sigma\mathbf{S}(1345)$

For $\Sigma\mathbf{S}(1345)$ to be useful we must be able to show its soundness relatively easily. I doubt that many would consider that showing the Negative Monotonicity Condition is relatively easy. To do so for a particular ϕ potentially involves using a great deal of knowledge about the structure of trees in $\Sigma\mathbf{S}(1345)$. In this section, I define a condition which uses less information about the structure of trees, but which implies the Negative Monotonicity Condition. I will call this condition "alphabetical monotonicity". The theoretical point about alphabetical monotonicity, which I will make in this section, is that it is a sufficient condition for the soundness of $\Sigma\mathbf{S}(1345)$. The pragmatic point, which I will make in Chapter 6, is that alphabetical monotonicity holds for a wide range of Σ-logics.

Notation

> *In the name "alphabetical monotonicity", the word "monotonicity" arises exactly as in the Negative Monotonicity Condition. I use the word "alphabetical" to emphasise that the condition uses names instead of the use of positions in the Negative Monotonicity Condition.*

I need to set things up for the definition of alphabetical monotonicity. This will involve making definitions that capture some features of trees. Having done this it will be possible to forget that we were talking about trees at all, at least for the purposes of proving alphabetical monotonicity.

Elements of $\mathbf{P}^{\delta_r}$ share two principal features. Firstly, they are all Σ-atoms. Secondly, each arises at the expansion of a δ-formula. That is, they arise in the expansion of $\mathbf{0}\forall x_{:\rho(x)}\,\phi$ or $\mathbf{1}\exists x_{:\rho(x)}\phi$. Now, there is nothing in the definition of Σ-logic to demand that x must appear in $\rho(x)$. However, for the purposes of alphabetical monotonicity I will demand this. This does not affect expressibility of Σ-logic at all, as the following argument shows. Suppose that $\rho(x)$ is an instance of a n place predicate not containing x. Then we can add to Σ an $n+1$ place predicate ρ', put x in the last argument position and replace $\rho(x)$ by $\rho'(x)$ in ϕ. If we add suitable clauses to Σ, expressing the equivalence of $\rho(x)$ and $\rho'(x)$, the Σ-validity of ϕ will be unchanged. The following clauses would work:

$$\rho(x) \rightarrow \rho'(x)$$

$$\rho'(x) \rightarrow \rho(x)$$

For alphabetical monotonicity, I wish to get away from considering trees. Instead, I will consider just sets of Σ-literals having certain properties. These properties will be closely modelled on the properties that sets of Σ-literals have in a tree.

Now I am assuming that at each δ-expansion, the introduced name appears in the introduced restriction. If we are considering a tree, we can look at which name is introduced. However, I wish to get away from considering trees. This means that I have to introduce some mechanism for deciding which names correspond to the introduced name. To do this, I will assume that, given a set of literals **R**, there is a function

$$new : \mathbf{R} \rightarrow Names$$

such that for any element ρ of **R**, $new(\rho)$ represents a new name in ρ. From my assumption, I may assume that $new(\rho)$ appears somewhere in ρ.

The next step is to observe that for some ϕ, it is not possible to introduce a new name into all argument positions in a Σ-predicate. For example, consider ϕ arising from the translation of modal logics into Σ-logic. The translation of $\mathbf{1}\Diamond\psi$ is, for some previously introduced name ε, $\mathbf{1}\exists x_{:\varepsilon\mathcal{R}x}\psi$. Similarly, the translation of $\mathbf{0}\Box\psi$ is $\mathbf{0}\forall x_{:\varepsilon\mathcal{R}x}\psi$. On reducing either of these by (δ), the introduced name can only arise in the second argument position of $\mathcal{R}$. I shall say that only the second argument position of $\mathcal{R}$ is *eligible*. For the purposes of alphabetical monotonicity, I will assume that each Σ-predicate has a certain number of (and certainly at least one) *eligible* argument positions.

Another concept I will use is that of *name-acyclicity*. By name acyclicity I mean that a set of restrictions can be ordered so that the new name in each restriction has not appeared in an earlier restriction. It is easy to show that sets of restrictions in $\Sigma\mathbf{S}(1345)$ proofs satisfy this condition. The reason I put this into the hypotheses of alphabetical monotonicity is that later I will define a stronger condition called "non-vacuity", and I will use name acyclicity to show that non-vacuity implies alphabetical monotonicity. The concept of name-acyclicity will have further importance when I discuss non-vacuity.

Definition 5.2.1 (name-acyclic)

A set of restrictions $\mathbf{R}$ is *name-acyclic* given a function $new : \mathbf{R} \to Names$ if $\mathbf{R}$ can be ordered as $\{ \rho_1, \rho_2, \ldots \}$ such that for each i, the name $new(\rho_i)$ does not appear in any any earlier restriction ρ_j with $j < i$.

Definition 5.2.2 (hypotheses of alphabetical monotonicity)

Sets of Σ-atoms $\mathbf{R}$, $\mathbf{R}'$, the function *new*, and the Σ-atom $\gamma_r(t)$, together satisfy the hypotheses of alphabetical monotonicity, iff:

(i) for each Σ-atom ρ in $\mathbf{R}$ and $\mathbf{R}'$, the name $new(\rho)$ appears in ρ in an eligible argument position; and

(ii) *new* does not map two different elements of $\mathbf{R}$, $\mathbf{R}'$, to the same name. That is, the function $new : \mathbf{R} \cup \mathbf{R}' \to Names$ is one to one; and

(iii) no name in $\mathbf{R}$ is a new name in $\mathbf{R}'$. That is, $names(\mathbf{R}) \cap new(\mathbf{R}') = \varnothing$; and

(iv) no name in $\gamma_r(t)$ is a new name in $\mathbf{R}'$. That is, $names(\gamma_r(t)) \cap new(\mathbf{R}') = \varnothing$; and

(v) the set $\mathbf{R} \cup \mathbf{R}'$ is name-acyclic given *new*.

Definition 5.2.3 (alphabetical monotonicity)

Σ *satisfies alphabetical monotonicity* iff,

for each set of Σ-atoms $\mathbf{R}$, $\mathbf{R}'$, function *new*, and Σ-atom $\gamma_r(t)$ which satisfy the hypotheses of alphabetical monotonicity:

if $\mathbf{R} \nvDash_\Sigma \gamma_r(t)$ then $\mathbf{R}, \mathbf{R}' \nvDash_\Sigma \gamma_r(t)$.

The reason for this somewhat complex definition is to make the condition as weak as possible while still mimicking the role that positions play in trees. This makes the following proof essentially trivial. However, the proof is still worth doing because I will show later that alphabetical monotonicity is not too difficult to demonstrate. In particular, if we can prove that alphabetical monotonicity holds without using all the hypotheses, we can safely ignore the extra ones. I will only need to use (v) when I introduce ‘‘non-vacuity’’, and show that it implies alphabetical monotonicity. Non-vacuity will often be extremely easy to show, and will also give rise to a better matrix characterisation of validity than alphabetical monotonicity. For the moment, though, back to the subject at hand.

Theorem 5.2.4: Soundness of $\Sigma\mathbf{S}(1345)$

If Σ satisfies alphabetical monotonicity, and if every δ-expansion in any $\Sigma\mathbf{S}(1345)$ tree of ϕ introduces a name only into an eligible position, then Σ satisfies the Negative Monotonicity Condition at ϕ.

Proof

Consider any expansion of γ to $\gamma_0(t)$ in a $\Sigma\mathbf{S}(1345)$ tree $\mathcal{T}$ for ϕ. If $\mathbf{R}$ is the set of restrictions on the branch at the point of expansion, I must show that if $\mathbf{R} \nvDash_\Sigma \gamma_r(t)$ then $\gamma_r(t) \notin LHM_\Sigma(\mathcal{T})$, or equivalently that $\mathbf{P}^{\delta_r} \nvDash_\Sigma \gamma_r(t)$. To do this I show how to satisfy the hypotheses of alphabetical monotonicity.

Define $\mathbf{R}' = \mathbf{P}^{\delta_r} - \mathbf{R}$, and the function *new* on $\mathbf{R}$, $\mathbf{R}'$, by that $new(\delta_r(a)) = a$.

Now, the sets **R** and **R′**, the function *new*, and the atom $\gamma_r(t)$ satisfy the hypotheses of alphabetical monotonicity, as the following arguments show:

(i) This is satisfied by the definition of *new* and the assumption in the statement of the theorem.

(ii) *new* is one-to-one since the condition (*4*) is satisfied in $\mathcal{T}$.

(iii) Any name in **R** must be introduced by a δ- or a γ-expansion on the branch. But any γ expansion only introduces ε or a name already on the branch, by (*3*). So any name in *names*(**R**) is either ε or must have been introduced by a δ-expansion on the branch. That is, $names(\mathbf{R}) \subseteq new(\mathbf{R}) \cup \{\varepsilon\}$. But as *new* is one-to-one, and **R** and **R′** are disjoint, $new(\mathbf{R}) \cap new(\mathbf{R}') = \varnothing$, as required.

(iv) Since (*3*) is satisfied, each name in $\gamma_r(t)$ appears in **R**.

(v) A simple ordering for $\delta_r(a) \in \mathbf{P}^{\delta_r}$ will do. We say that δ_r^1 is earlier in the ordering than δ_r^2 if either δ_r^1 appears on a more leftmost branch than δ_r^2, or if their leftmost appearance is on the same branch but δ_r^1 appears higher on that branch. Now consider the first occurrence of some restriction $\delta_r^2 \in \mathbf{P}^{\delta_r}$ containing a name *a* associated with a different position δ_r^1. The name *a* can only arise in δ_r^2 by an earlier γ or δ expansion on the branch δ_r^2 appears in, since (*4*) is satisfied. Furthermore, if it arises because of a γ-expansion then it must be introduced by the introduction of a δ-expansion since (*3*) is satisfied. In either case the δ-expansion introducing $\delta_r^1(a)$ must appear before δ_r^2 on the branch. This argument applies in particular to the leftmost occurrence of δ_r^2, and so δ_r^1 is earlier in the ordering than δ_r^2. This applies to any positions δ_r^1 and δ_r^2, and so implies that the set of restrictions is indeed name-acyclic.

As the hypotheses of alphabetical monotonicity are satisfied, and Σ is assumed to satisfy alphabetical monotonicity, we know that if $\mathbf{R} \nvDash_\Sigma \gamma_r(t)$ then $\mathbf{R}, \mathbf{R}' \nvDash_\Sigma \gamma_r(t)$. By definition, $\mathbf{R} \cup \mathbf{R}' = \mathbf{P}^{\delta_r}$, and so alphabetical monotonicity gives us exactly what I needed to show.

□

Section 5.3: Non-Vacuity and the Soundness of ΣS(*145*)

Alphabetical monotonicity ensures that (*5*) is in fact equivalent to (*2*) whenever a (γ)-expansion is applied. However, it requires that the condition (*3*) hold everywhere in a tree. Later, when I develop the matrix characterisation of validity, this requirement will cause problems. Therefore, I now show how a stronger condition on Σ can be sufficient for the soundness of an expansion system containing (*5*) but not (*3*).

Recall that the expansion system ΣS(*145*) is the expansion system containing the propositional expansion rules from **S**, the rules (γ) and (δ), and the side conditions (*1*), (*4*), and (*5*). The completeness of ΣS(*145*) is trivial, since we have that ΣS(*124*) is complete, and any ΣS(*124*)-tree is also a ΣS(*145*)-tree.

Theorem 5.3.1: Completeness of ΣS(*145*)

If ϕ *is a restricted* Σ*-formula for a theory* Σ *that satisfies the least Herbrand model property, then:*

if ϕ *is* Σ*-valid then there is a closed* ΣS(*145*) *tree for* ϕ.

□

Soundness is not at all trivial. Indeed, soundness fails for the *empty* theory!

Example (Failure of soundness of* ΣS(*145*)*)

Suppose Σ is empty, and that a ρ is a one-argument Σ-predicate. Consider the signed formula

$$\mathbf{1}\ (\forall x_{:\rho(x)}\ \mathbf{f}) \wedge (\ (\exists y_{:\rho(y)}\ \mathbf{t}) \vee \mathbf{t})$$

This formula is certainly satisfiable. It is satisfied by any Σ-model in which ρ is false of everything. We can look for a closed ΣTab(*124*) tree for this formula, but we cannot find one. If we try, we come up with:

$$
\begin{array}{c}
\underline{\mathbf{1}\ (\forall x_{:\rho(x)}\ \mathbf{f}) \wedge (\ (\exists y_{:\rho(y)}\ \mathbf{t}) \vee \mathbf{t})} \\
\mathbf{1}\ (\forall x_{:\rho(x)}\ \mathbf{f}) \\
\mathbf{1}(\exists y_{:\rho(y)}\ \mathbf{t}) \vee \mathbf{t} \\
\hline
\begin{array}{c}
\underline{\mathbf{1}\ \exists y_{:\rho(y)}\ \mathbf{t}} \\
\underline{\mathbf{1}\ \mathbf{t}} \\
\underline{\rho(a)} \\
\underline{\mathbf{1}\ \mathbf{f}} \\
\otimes
\end{array}
\quad
\mathbf{1}\ \mathbf{t}
\end{array}
$$

The introduction of **1f** on the left hand branch satisfies (*2*) because $\rho(a)$ is present on the branch.

This tree is not expandable, because on the right hand branch, for no term t (including the name ε) does $\varnothing \vDash_{\Sigma} \rho(t)$, so no possible γ-expansion can satisfy the condition (*2*) in Σ**S**(*12*). The tree is not closed. Indeed this is the only Σ**Tab**(*124*) tree there is for the original formula.

If we allow ourselves to introduce positions that are in $LHM_{\Sigma}(\mathcal{T})$, we can extend the above tree to produce a closed Σ**Tab**(*145*) tree for the original formula:

$$
\begin{array}{c}
\underline{\mathbf{1}\ (\forall x_{:\rho(x)}\ \mathbf{f}) \wedge (\ (\exists y_{:\rho(y)}\ \mathbf{t}) \vee \mathbf{t})} \\
\mathbf{1}\ (\forall x_{:\rho(x)}\ \mathbf{f}) \\
\mathbf{1}(\exists y_{:\rho(y)}\ \mathbf{t}) \vee \mathbf{t} \\
\hline
\begin{array}{c}
\underline{\mathbf{1}\ \exists y_{:\rho(y)}\ \mathbf{t}} \\
\underline{\mathbf{1}\ \mathbf{t}} \\
\underline{\rho(a)} \\
\underline{\mathbf{1}\ \mathbf{f}} \\
\otimes
\end{array}
\quad
\begin{array}{c}
\underline{\mathbf{1}\ \mathbf{t}} \\
\underline{\mathbf{1}\ \mathbf{f}} \\
\otimes
\end{array}
\end{array}
$$

The introduction of **1f** on the right hand branch satisfies (*5*) because $\rho(a)$ is in $LHM_{\Sigma}(\mathcal{T})$.

End of Example

So Σ**Tab**(*145*) is unsound and so in general Σ**S**(*145*) can be unsound. The next question is, can we characterise a range of cases when Σ**S**(*145*) is in fact sound?

I will now present a condition on Σ that ensures that Σ**S**(*145*) is sound. This is done by showing that if the condition is satisfied and there is a closed Σ**S**(*145*)-tree for a set of formulas, then there is a closed Σ**S**(*124*)-tree for that set of formulas. The result follows from the soundness of Σ**S**(*124*). The next chapter contains examples of proving that the condition holds.

Notation

> *I call the condition ''non-vacuity''. This is because the problem revealed above was that an empty domain was possible on one branch, but not on another. By ensuring that an empty domain is never possible, the problem disappears. The actual condition generalises this idea.*

Non-vacuity is a much simpler condition to state than alphabetical monotonicity, and this is reflected in the fact that it is much easier to prove (where true) as I will show in Chapter 6.

As for alphabetical monotonicity, I will use the notion of *eligible* argument positions in Σ-predicates. Now I will assume that each Σ-predicate has exactly one eligible argument position. Earlier I assumed that each Σ-predicate had at least one eligible position. All the logics I consider in this thesis satisfy this assumption, and makes the definitions and proofs easier. One could easily do very similar work without the assumption.

Notation

> *I will write* $\rho(\underline{t};t_0)$ *to indicate that* ρ *takes the vector of arguments* $\underline{t}$ *in non-eligible positions and the argument* t_0 *in its eligible position.*

Definition 5.3.2 (Non-vacuity)

Σ satisfies non-vacuity iff,

for each Σ-predicate ρ and for each vector of arguments $\underline{t}$, there is some term t_0 such that, with the possible exception of ε, all the names in t_0 appear in $\underline{t}$, and such that $\varnothing \vDash_\Sigma \rho(\underline{t};t_0)$.

The empty theory (used in the example above) is a particularly simple case where non-vacuity fails to hold. With nothing in Σ we certainly cannot prove that $\varnothing \vDash_\Sigma \rho(\underline{t};t_0)$. More complex theories failing non-vacuity arise in modal logics. In particular, logics which are non-reflexive and non-serial fail it. So in the theory containing just the clause for transitivity, namely

$$(w\mathcal{R}x \wedge x\mathcal{R}y) \rightarrow w\mathcal{R}y,$$

we cannot prove $\varnothing \vDash_\Sigma t\mathcal{R}t_0$ for any t or t_0.

Modal theories containing seriality satisfy non-vacuity. The clause for seriality is:

$$\rightarrow w\mathcal{R}next(w).$$

Non vacuity holds since the only eligible argument position in $\mathcal{R}$ is the second one, and for any t we can show $\varnothing \vDash_\Sigma t\mathcal{R}\,next(t)$. Further examples where non-vacuity holds are given in Chapter 6.

I show, in two steps, that Σ**S**(*145*) is sound if non-vacuity holds. First I will show that if Σ satisfies non-vacuity then it must also satisfy alphabetical monotonicity. Then I will use this result to show the soundness of the expansion rule (γ) given the side condition (*5*). However, this soundness will rely on an extra condition: that the restrictions in a proof are name-acyclic. It is unpleasant to have to enforce this extra condition. Fortunately, however, the most important use I make of the soundness of Σ**S**(*145*) is in the derivation of a matrix characterisation of validity (in Chapter 7): in that context name-acyclicity is guaranteed by conditions that are essential for other reasons. I first need to make a simple observation.

Notation

When dealing with assignments, I will write "$t\sigma$" or "$(t)\sigma$" to indicate the application of the assignment σ to the term t. If σ_1 and σ_2 are both assignments, then "$\sigma_1\sigma_2$" indicates the assignment resulting if σ_1 is applied followed by σ_2. Finally, I will write, for example, $[a_1 := t_1, a_2 := t_2]$ to indicate the assignment in which a_1 is replaced by t_1 and a_2 is replaced with t_2.

Observation 5.3.3

For any set **R** of Σ-atoms, Σ-atom ρ, and assignment σ of names to terms, if

$$\mathbf{R} \vDash_\Sigma \rho$$

then

$$(\mathbf{R})\sigma \vDash_\Sigma (\rho)\sigma$$

Proof

An easy way to see this is to use the fact that Σ is expressed in ordinary first order logic. By substitutivity, if $\mathbf{R} \vDash_\Sigma \rho$ then $(\mathbf{R} \vDash_\Sigma \rho)\sigma$. But by definition, Σ contains no *names*. Therefore $(\Sigma)\sigma = \Sigma$, giving the desired result.

□

The next lemma illustrates the crucial use of my earlier definition of a set of restrictions being name-acyclic. The rest of the proofs in this section will rely on this lemma.

Lemma 5.3.4

If Σ satisfies non-vacuity, and $\mathbf{R}$ and $\mathbf{R}'$ are sets of restrictions with associated function *new* such that $\mathbf{R} \cup \mathbf{R}'$ is name-acyclic, then there is an assignment σ of new names in $\mathbf{R}'$ to terms built up from the new names in $\mathbf{R}$, constants, and the name ε, such that

$$(\mathbf{R})\,\sigma \vDash_\Sigma (\mathbf{R}')\,\sigma$$

Proof

Firstly, consider any names in $\mathbf{R}$ and $\mathbf{R}'$ that do not appear as new names somewhere. Any such names play no special role and we may assign them arbitrarily, for instance to the special name ε. Let σ_0 be the assignment assigning all such names to ε (or the empty assignment if there are no such names).

Because the set is name-acyclic, we may order $\mathbf{R} \cup \mathbf{R}'$ as

$$\rho_1(\underline{t}_1\,; a_1),\ \rho_2(\underline{t}_2\,; a_2), \ldots$$

where each $\underline{t}_i$ is a vector of terms and each a_i is a single name $new(\rho_i)$ such that a_i does not appear in any t_j for $j \le i$.

We build up an assignment σ from $\sigma_0, \sigma_1, \sigma_2, \ldots$ We start with σ_0, as defined in the first paragraph. We construct σ_{i+1} from σ_i as follows.

If $\rho_{i+1}(\underline{t}_{i+1}\,; a_{i+1}) \in \mathbf{R}$ then we set $\sigma_{i+1} = \sigma_i$.

Otherwise $\rho_{i+1}(\underline{t}_{i+1}\,; a_{i+1}) \in \mathbf{R}'$. Because the set of restrictions is name-acyclic a_{i+1} is new, and by construction σ_i does not assign anything to a_{i+1}. So $(\rho_{i+1}(\underline{t}_{i+1}\,; a_{i+1}))\sigma_i = \rho_{i+1}((\underline{t}_{i+1})\,\sigma_i\,; a_{i+1})$. By non-vacuity, there is some term t_{i+1}, containing only names in $(\underline{t}_{i+1})\sigma_i$ or ε, such that

$$\varnothing \vDash_\Sigma \rho_{i+1}((\underline{t}_{i+1})\sigma_i\,; t_{i+1}) = (\rho_{i+1}(\underline{t}\,; a_{i+1})\,\sigma_i)\,[a_{i+1} := t_{i+1}]$$

In this case set $\sigma_{i+1} = \sigma_i\,[a_{i+1} := t_{i+1}]$.

The assignments σ_i can be combined into a single assignment $\sigma = [a_1 := t_1, a_2 := t_2, \ldots]$ so that $(a_i)\,\sigma = t_i$. We have that if $\rho_i \in \mathbf{R}'$ then $\varnothing \vDash_\Sigma (\rho_i)\sigma$, so

$$(\mathbf{R})\,\sigma \vDash_\Sigma (\mathbf{R}')\,\sigma$$

□

Corollary 5.3.5: Non-vacuity implies alphabetical monotonicity

If Σ satisfies non-vacuity then it satisfies alphabetical monotonicity.

Proof

Consider any sets of Σ-atoms $\mathbf{R}$ and $\mathbf{R}'$, function *new*, and Σ-atom $\gamma_r(t)$ satisfying the hypotheses of alphabetical monotonicity, and additionally satisfying my assumption that each Σ-predicate has exactly one eligible argument position. From part (v) of these hypotheses, $\mathbf{R} \cup \mathbf{R}'$ is name-acyclic. From Lemma 5.3.4 there is an assignment σ such that

$$(\mathbf{R})\,\sigma \vDash_\Sigma (\mathbf{R}')\,\sigma. \tag{1}$$

Suppose that $\mathbf{R}, \mathbf{R}' \vDash_\Sigma \gamma_r(t)$. Then by Observation 5.3.3

$$(\mathbf{R}, \mathbf{R}')\sigma \vDash_\Sigma (\gamma_r(t))\sigma \tag{2}$$

(1) and (2) together imply that

$$(\mathbf{R})\sigma \vDash_\Sigma (\gamma_r(t))\sigma$$

But by the hypotheses of alphabetical monotonicity, and since each $a_i = new(\rho_i)$, $\mathbf{R}$ and $\gamma_r(t)$ contain no

occurrence of any a_i if $\rho_i \in \mathbf{R}'$. Therefore $(\mathbf{R})\sigma = \mathbf{R}$ and $(\gamma_r(t))\sigma = \gamma_r(t)$. Hence, $\mathbf{R} \vDash_\Sigma \gamma_r(t)$ and Σ satisfies alphabetical monotonicity.

□

Lemma 5.3.4 is certainly false for a set of restrictions that are not name acyclic. For example, the modal theory containing only seriality satisfies non-vacuity (as I show in Chapter 6). Consider the two sets of restrictions $\mathbf{R} = \varnothing$ and $\mathbf{R}' = \{ a \mathcal{R} b, b \mathcal{R} a \}$. Although from seriality we have that $\varnothing \vDash_\Sigma a \mathcal{R} next(a)$ and $\varnothing \vDash_\Sigma b \mathcal{R} next(b)$, no assignment σ can satisfy both restrictions simultaneously. Although the main theorem of this section, Theorem 5.3.7, relies on Lemma 5.3.4, I have not yet been able to establish that it is necessary, as well as sufficient for the theorem. That is, I do not have an example where allowing non name-acyclic proofs allows the construction of closed trees for invalid formulas.

The soundness proof I am about to give is based on viewing a branch in a $\Sigma\mathbf{S}(145)$ proof as a disguised version (by some assignment σ) of a branch in a $\Sigma\mathbf{S}(12)$ proof, which we know to be sound from Chapter 4. To this end the following observation is crucial.

Observation 5.3.6

If S is a set of Σ-formulas and σ is an assignment of names to terms, then S is Σ-satisfiable if $(S)\sigma$ is.

Proof

Say $(S)\sigma$ is satisfied by a Σ-model $\mathbf{M} = <\mathbf{D},\mathbf{I}>$. We can base a new interpretation function $\mathbf{J}$ on $\mathbf{I}$ to provide a model for S. Simply define $\mathbf{J}$ by:

for names a, $a^{\mathbf{J}} = [\,(a)\sigma\,]^{\mathbf{I}}$; and

for functions f, $f^{\mathbf{J}}$ is defined by $[f(t_1, \ldots, t_n)]^{\mathbf{J}} = [\,(f(t_1, \ldots, t_n))\,\sigma\,]^{\mathbf{I}}$; and

for predicates P, $P^{\mathbf{J}}$ is defined by $[P(t_1, \ldots, t_n)]^{\mathbf{J}} = [\,(P(t_1, \ldots, t_n))\,\sigma\,]^{\mathbf{I}}$.

Then the model $< \mathbf{D}, \mathbf{J} >$ serves as a model for S.

□

Theorem 5.3.7: Soundness of $\Sigma\mathbf{S}(145)$ given non-vacuity

If **S** *is a sound propositional expansion system, Σ satisfies non-vacuity, and there is a closed $\Sigma\mathbf{S}(145)$ tree for S in which the set of restrictions introduced by δ-expansions is name-acyclic, then S is Σ-unsatisfiable.*

Proof

I prove the contrapositive: that if S is Σ-satisfiable there is no closed tree for S.

Suppose that $\mathcal{T}$ is a $\Sigma\mathbf{S}(145)$ tree for S. Then for each branch λ let $\mathbf{R}_\lambda$ be the set of restrictions introduced by δ-expansions on the branch. By Lemma 5.3.4, and the assumption that $\mathbf{P}^{\delta_r}$ is name-acyclic, for each branch λ there is an assignment σ_λ from names not introduced by δ-expansions on λ such that:

$$(\mathbf{R}_\lambda)\,\sigma_\lambda \vDash_\Sigma (\mathbf{P}^{\delta_r})\,\sigma_\lambda \qquad (1)$$

If λ and μ are different branches we can demand that the assignments σ_λ and σ_μ agree on names appearing on the common part of the two branches. To see this, note that from the proof of Lemma 5.3.4 to construct σ_λ we need to find terms to assign to each name a introduced by δ-expansions on different branches. In each case we choose names such that $\varnothing \vDash_\Sigma (\rho(a))\sigma_\lambda$. But if a name appears on the common part of two branches and is assigned by either σ_λ or σ_μ, then that name is not introduced by a δ-expansion on the common part of λ and μ. But since the name appears on the common part of the branch, which is above the separate parts, and since each δ-expansion satisfies (1), no expansion on either branch can introduce that name. So we can construct both σ_λ and σ_μ by descending the branch, choosing a suitable assignment for each name jointly until the two branches separate.

Now we can look at each branch λ under the assignment σ_λ. I show that if U is the set of formulas up to a certain point on the branch λ, then the application of the expansion rule, viewed under the assignment σ_λ

is a sound one. Reexpressing this precisely: if $(U)\sigma_\lambda$ is Σ-satisfiable, $\{X_1, \ldots, X_n\} \subseteq U$, and

$$\frac{\begin{array}{c} X_1 \\ \cdot \\ \cdot \\ X_n \end{array}}{Y_1 \mid \cdots \mid Y_m}$$

is an application of an expansion rule, then
for some i, $(U \cup \{Y_i\})\sigma_\lambda$ is Σ-satisfiable.

First note that the expansion viewed under σ is a correct expansion of $(U)\sigma_\lambda$, if we ignore side conditions. The only case to check is that of a branching propositional rule. But a propositional rule introduces no new names, and for any branch λ_i descended from Y_i, $(Y_i)\sigma_{\lambda_i} = (Y_i)\sigma_\lambda$ because of the earlier demand that each σ_λ is identical on names in a shared part of a branch.

The second task is to show that the expansion viewed under σ is a sound one. Since propositional rules are sound without side conditions, we are left only with quantifier rules. Also, the new name introduced by a δ-expansion will be unchanged by the assignment σ_λ and will be new on the branch. Therefore any δ-application will satisfy (*1*) and so is sound (by Theorem 4.3.2).

The tricky case is that of a γ-expansion. Since the original γ-application must satisfy (*5*), $\mathbf{P}^{\delta_r} \vDash_\Sigma \gamma_r(t)$. From (1), $(\mathbf{R}_\lambda)\sigma_\lambda \vDash_\Sigma (\gamma_r(t))\sigma_\lambda$. Now, $\mathbf{R}_\lambda$ is the set of restrictions on the whole branch λ, which can be a superset of the set of restrictions $\mathbf{R}_U$ contained in U. But $(\mathbf{R}_U)\sigma_\lambda$, $(\mathbf{R}_\lambda - \mathbf{R}_U)\sigma_\lambda$, *new* defined in the usual way, and $(\gamma_r(t))\sigma_\lambda$ satisfy the hypotheses of alphabetical monotonicity. Σ satisfies alphabetical monotonicity because it satisfies non-vacuity (Corollary 5.3.5). So by alphabetical monotonicity, $(\mathbf{R}_U)\sigma_\lambda \vDash_\Sigma (\gamma_r(t))\sigma_\lambda$. Therefore the (γ) application viewed under the assignment σ_λ in fact satisfies the side condition (*2*), and so is a sound application as shown by Theorem 4.3.1.

Now consider the tree $\mathcal{T}$, whose root is the set S containing no names. Then for any assignment, $S = (S)\sigma$. Then for some branch λ and associated assignment σ_λ and set of formulas U, by the soundness of the expansion rules under σ_λ, since $(S)\sigma_\lambda = S$ is Σ-satisfiable, $(U)\sigma_\lambda$ is Σ-satisfiable, and by Observation 5.3.6, U is Σ-satisfiable and so cannot be closed. So if S is Σ-satisfiable there is no closed **ΣS**(*145*) tree for S.

□

Although I have shown that **ΣS**(*145*) can be sound, this is at the expense of demanding name-acyclicity. This is a global condition on proof trees, rather like (*4*). In later proof systems the condition will be subsumed by other conditions. That is, the usual problem of ensuring the soundness of existential instantiation will arise: in solving this, for example as I do in Chapter 7 with Bibel's reduction orderings, one ensures name-acyclicity anyway.

Chapter 6

Proving Alphabetical Monotonicity and Non-vacuity

I have shown in Chapter 5 how alphabetical monotonicity and non-vacuity justify the soundness of $\Sigma\mathbf{S}(1345)$ and $\Sigma\mathbf{S}(145)$ respectively. In this chapter I discuss how these conditions can be proved for a given theory Σ. I discuss non-vacuity in Section 6.1 and alphabetical monotonicity in Section 6.2. As alphabetical monotonicity is implied by non-vacuity (Corollary 5.3.5), it is only necessary to consider theories where non-vacuity fails.

The proofs of non-vacuity in this chapter are very much easier than those of alphabetical monotonicity. This is interesting. Non-vacuity seems to win on all counts over alphabetical monotonicity: it is easier to prove; it implies alphabetical monotonicity anyway; and when it is satisfied we can throw away (3). It is this last property that is most important, playing a significant role in matrix characterisations of validity (in Chapter 7), where a nicer matrix system is possible than where only alphabetical monotonicity holds, and being essential for instantiation rules (in Chapter 8). It is not obvious to me why this important condition should be so easy to prove.

Section 6.1: Proving Non-vacuity

If we can prove non-vacuity for a theory Σ, then the proof system $\Sigma\mathbf{S}(145)$ is sound[1], and this is a property we are very interested in. Fortunately, non-vacuity is often a very easy property to prove. In this section I show how to prove it for serial modal logics and for sorted logics with non-empty domains. These proofs will be very easy, contrasting with the proofs of alphabetical monotonicity I give later in the Chapter.

For convenience I repeat the definition of non-vacuity.

Definition 5.3.2 (Non-vacuity)

Σ satisfies non-vacuity iff,

for each Σ-predicate ρ and for each vector of arguments $\underline{t}$, there is some term t_0 such that, with the possible exception of ε, all the names in t_0 appear in $\underline{t}$, and such that $\varnothing \vDash_\Sigma \rho(\underline{t};t_0)$.

Section 6.1.1: Non-vacuity in Sorted Logics

Consider non-vacuity for sortal logics in Frisch's (1989) framework. Frisch's syntax and semantics are essentially the same as mine, except that all Σ-predicates must be monadic (that is, of one argument). This includes, for example, order-sorted logics. Where Σ-predicates are monadic, non-vacuity becomes a triviality. That is, in all cases ρ takes only one argument so the quantification over vectors of arguments $\underline{t}$ is empty. Non-vacuity is equivalent to the demand:

for each Σ-predicate ρ, there is some term t such that $\varnothing \vDash_\Sigma \rho(t)$.

Rephrasing this in the language of sorts, one derives the following theorem.

[1] As I mentioned in Chapter 5, I am ignoring the problem of ensuring name-acyclicity. This is unnecessary in Chapter 7 where it is subsumed by another check used there.

Theorem 6.1.1.1: Non-vacuity for Sorted Logics with Non-empty Domains

If a sorted logic contains sorts $s_1, s_2, \ldots s_n$ and a sort theory Σ, then the sort theory satisfies non-vacuity iff the sort theory ensures that the domain of each sort s_i is non-empty.

□

Section 6.1.2: Non-vacuity in Serial Modal Logics

Serial and reflexive modal logics satisfy non-vacuity. For the first order variants, we must demand that the domain in each world is non-empty. The proof of these facts is extremely simple, and I give it below. It is interesting to note the extreme ease of these proofs compared to the difficulty of proofs of alphabetical monotonicity for modal logics, yet these proofs cover a wider range of logics. Because of these results, we need only consider alphabetical monotonicity in non-serial modal logics and in this thesis the only such modal logics I have considered are **K**, **K4** and **KB**.

Theorem 6.1.2.1: Non-vacuity for Serial Modal Logics with Non-empty Domains

All the modal theories containing either reflexivity or seriality, and which ensure that the domain of each world is non-empty, satisfy non-vacuity.

Proof

For modal theories we assume that only the second argument position of $\mathcal{R}$ is eligible, and the first argument position of *in*. I will show that non-vacuity holds by considering first *in* and then $\mathcal{R}$.

If the theory Σ ensures that the domain of each world is non-empty, then for any world w there is a term $t(w)$ such that $\varnothing \vDash_\Sigma t(w)\ in\ w$.

If the theory Σ contains reflexivity, then for any world w we have that $\varnothing \vDash_\Sigma w\ \mathcal{R}\ w$. If Σ contains seriality, then instead we have that $\varnothing \vDash_\Sigma w\ \mathcal{R}\ next(w)$. In either case we have some term as required by non-vacuity.

□

Section 6.2: Proving Alphabetical Monotonicity

The intention of my work is to make it as easy as possible to establish the correctness of a variety of interesting proof systems. For instance, now we can read off the soundness and completeness of $\Sigma\mathbf{S}(1345)$ for any countermodel expansion system **S**, if the relevant theory satisfies alphabetical monotonicity and the least Herbrand model property. In Chapter 7 this result will lead to a matrix characterisation of validity. It is trivial to observe that a theory is expressible in Horn clauses and therefore satisfies the least Herbrand model property. Unfortunately, proving alphabetical monotonicity for a given theory is not trivial. So it is desirable to have a further range of results which give easy conditions to check if a theory satisfies alphabetical monotonicity. Any such result must rest on the *global* properties of the theory: it is not enough, for instance, that each clause satisfies alphabetical monotonicity as their combinations may not do so. For instance, the combination of transitivity and symmetry fails alphabetical monotonicity (see Section 5.3) while each on its own satisfies it.

I can see no general way of deciding alphabetical monotonicity. So in this section I demonstrate that a range of theories satisfy alphabetical monotonicity. I am able to give one general result which shows that alphabetical monotonicity holds if the clauses satisfy certain properties. This result, which I call the "Marking Theorem" takes advantage of common features of proofs of alphabetical monotonicity, meaning that a comparatively simple check is all that is required for a particular theory. A simple corollary of the Marking Theorem gives a simply checked syntactic property which is sufficient for alphabetical monotonicity. I call this property "Reverse Range Restriction". Reverse range restriction holds in the theories of some modal logics and in a large number of sorted logics, including order sorted logics. Although the Marking Theorem only provides a sufficient condition to satisfy alphabetical

monotonicity, not a necessary one, it is enough for all the theories I consider in this section.

In Section 6.2.1 I discuss the general nature of proofs of alphabetical monotonicity. In Section 6.2.2 I show that alphabetical monotonicity holds for the empty theory. In Section 6.2.3 I introduce reverse range restriction, although I leave the proof to the next section. I use reverse range restriction for the modal theory of symmetry and order sorted logics. In Section 6.2.4 I prove the Marking Theorem (and reverse range restriction as a corollary) and apply it in Section 6.2.5 to first order variants of the modal theory of transitivity.

Section 6.2.1: The Nature of Proofs of Alphabetical Monotonicity

Before proceeding, it is convenient to recall the definition of alphabetical monotonicity.

Definition 5.2.1 (name-acyclic)

A set of restrictions **R** is *name-acyclic* given a function $new : \mathbf{R} \rightarrow Names$ if **R** can be ordered as $\{ \rho_1, \rho_2, \ldots \}$ such that for each i, the name $new(\rho_i)$ does not appear in any any earlier restriction ρ_j with $j < i$.

Definition 5.2.2 (hypotheses of alphabetical monotonicity)

Sets of Σ-atoms **R**, **R′**, the function *new*, and the Σ-atom $\gamma_r(t)$, together satisfy the hypotheses of alphabetical monotonicity, iff:

(i) for each Σ-atom ρ in **R** and **R′**, the name $new(\rho)$ appears in ρ in an eligible argument position; and

(ii) *new* does not map two different elements of **R**, **R′**, to the same name. That is, the function $new : \mathbf{R} \cup \mathbf{R}' \rightarrow Names$ is one to one; and

(iii) no name in **R** is a new name in **R′**. That is, $names(\mathbf{R}) \cap new(\mathbf{R}') = \varnothing$; and

(iv) no name in $\gamma_r(t)$ is a new name in **R′**. That is, $names(\gamma_r(t)) \cap new(\mathbf{R}') = \varnothing$; and

(v) the set $\mathbf{R} \cup \mathbf{R}'$ is name-acyclic given *new*.

Definition 5.2.3 (alphabetical monotonicity)

Σ *satisfies alphabetical monotonicity* iff,

for each set of Σ-atoms **R**, **R′**, function *new*, and Σ-atom $\gamma_r(t)$ which satisfy the hypotheses of alphabetical monotonicity:

if $\mathbf{R} \nvDash_\Sigma \gamma_r(t)$ then $\mathbf{R}, \mathbf{R}' \nvDash_\Sigma \gamma_r(t)$.

As well as being convenient, there is a pedagogical purpose behind recalling these definitions. From the completeness of $\Sigma\mathbf{S}(1345)$ (Section 4.4.2), the soundness of $\Sigma\mathbf{S}(1345)$ given the Negative Monotonicity Condition (Theorem 5.1.4), and particularly the fact that alphabetical monotonicity implies the Negative Monotonicity Condition (Theorem 5.2.4), we know that if we can prove alphabetical monotonicity for a particular restriction theory Σ, we can use $\Sigma\mathbf{S}(1345)$ for that Σ. However, looking at the definitions above, to prove alphabetical monotonicity for Σ, we need know nothing about $\Sigma\mathbf{S}(1345)$ at all. Where it is possible, it is much simpler to prove alphabetical monotonicity, than to prove $\Sigma\mathbf{S}(1345)$ correct directly.

In the proofs of alphabetical monotonicity in this section, I will not in fact need to use part (v) of the hypotheses at all. Part (v) was necessary in the proof that non-vacuity implies alphabetical monotonicity (Corollary 5.3.5), but in all the examples in this section, parts (i) to (iv) will be strong enough for the result without (v).

I will use UR-resolution (Unit Resulting resolution) as a tool for proving alphabetical monotonicity. UR-resolution is complete for sets of Horn clauses (stated, for example in Eisinger and Ohlbach 1989)[2],

[2] Wos (1988) cites (McCharen *et al* 1976) as the original source for UR-resolution.

and I will only consider theories in Horn clause form.

In the context of Horn clauses, UR-resolution is a very natural procedure. The idea is simply that to prove ρ from **R** given Σ, one uses clauses in Σ in a forwards direction. That is, given a clause $B_1, \ldots B_k \rightarrow H$, one concludes H if we already know each B_i, either because it is in **R** or because we have already proved it. The only complication is that rather than just matching B_i against a literal in **R**, we must *unify* it. Anyone with any familiarity either with PROLOG or resolution theorem proving will know what this involves, and I will assume such familiarity with unification throughout this chapter.

The use I make of UR-resolution is as follows. I will consider a hypothetical UR-resolution proof of $\mathbf{R},\mathbf{R}' \vDash_\Sigma \rho$, which will be list of deductions of the following form, each based on a clause from Σ:

$$B^1_1, B^1_2, \ldots B^1_{m_1} \rightarrow H^1$$

$$B^2_1, B^2_2, \ldots B^2_{m_2} \rightarrow H^2$$

$$B^3_1, B^3_2, \ldots B^3_{m_3} \rightarrow H^3$$

$$\ldots$$

$$B^n_1, B^n_2, \ldots B^n_{m_n} \rightarrow H^n$$

where each B^i_j has been unified either with an element of $\mathbf{R},\mathbf{R}'$ or an earlier conclusion H^k, and where H^n has been unified with ρ. (Note that m_i is allowed to be 0, i.e. clauses containing only a head and no body are allowed.)

Given such a proof, I will be able to show that there is another one in which no element of $\mathbf{R}'$ is used. (I will do this by induction on the length n of the proof.)

Section 6.2.2: Alphabetical Monotonicity for Empty Theories

The easiest theories to prove alphabetical monotonicity for are *empty* theories.

Theorem 6.2.2.1: Alphabetical monotonicity for empty theories

If Σ is empty then Σ satisfies the least Herbrand model property and alphabetical monotonicity.

Proof

Since Σ is empty it is trivially in Horn clause form and hence satisfies the least Herbrand model property. Now suppose that we have **R**, $\mathbf{R}'$, *new*, and γ_r satisfying the hypotheses of alphabetical monotonicity.

Since Σ is empty and **R** and $\mathbf{R}'$ contain only atoms:

$$\mathbf{R}, \mathbf{R}' \vDash_\Sigma \gamma_r \text{ iff } \gamma_r \in \mathbf{R} \cup \mathbf{R}'$$

But if the hypotheses for alphabetical monotonicity are satisfied then $new(\mathbf{R}') \cap names(\gamma_r) = \varnothing$ and so $\gamma_r \notin \mathbf{R}'$. So $\gamma_r \in \mathbf{R} \cup \mathbf{R}'$ if and only if $\gamma_r \in \mathbf{R}$. This gives us, as required by alphabetical monotonicity, that

$$\mathbf{R}, \mathbf{R}' \vDash_\Sigma \gamma_r \text{ iff } \mathbf{R} \vDash_\Sigma \gamma_r$$

□

Although Σ being empty is a trivial property, non-trivial logics have empty theories. For example, in the modal logic **K** no demands are made on the accessibility relation, and hence when we translate **K** into a Σ-logic, Σ is empty. Also, in simple sorted logics (discussed in Section 2.6), the only syntactic extension to predicate logic is the attachment of sorts to variables. Unless we do demand that some sorts must be non-empty, the restriction theory will be empty. (If we demand that all sorts are non-empty then non-vacuity will be satisfied, as discussed in Section 6.1.)

Even where, as here, the proofs of alphabetical monotonicity are easy, the resulting proof systems still allow us to take full advantage of the parts of the semantics of the logic that are easy to reason in. Indeed, in the case of the empty theory, of course those parts of the logic are trivial to reason in.

Section 6.2.3: Reverse Range Restriction

In this section I give a sufficient condition for alphabetical monotonicity which is easy to check. There is a tradeoff between ease of checking sufficient conditions and the number of cases that the conditions cover. At one extreme is the trivial proof in the previous section for empty theories, but of course not many theories are empty. At the other extreme is the full definition of alphabetical monotonicity, which trivially covers all cases but is difficult to check. In this section I give a first intermediate point along this continuum, which can be characterised syntactically very easily. This is enough to prove that the modal theory for symmetry satisfies alphabetical monotonicity, as well as the theories for many sorted logics. In this section I will omit the proof that this condition does imply alphabetical monotonicity, since I prove it as an easy corollary of a more general condition that will cover more theories but be less easy to check in general.

The condition I introduce in this section is called "Reverse Range Restriction". It applies to theories Σ expressed as a list of Horn clauses (where each m_i may be 0):

$$\mathbf{C}^1 : B^1_1, B^1_2, \ldots B^1_{m_1} \rightarrow H^1$$

$$\mathbf{C}^2 : B^2_1, B^2_2, \ldots B^2_{m_2} \rightarrow H^2$$

$$\ldots$$

$$\mathbf{C}^n : B^n_1, B^n_2, \ldots B^n_{m_n} \rightarrow H^n$$

Definition 6.2.3.1 (Reverse Range Restricted)

A clause $\mathbf{C}^i$ is *reverse range restricted* if each *variable* that appears in any B^i_j also appears in H^i.

A theory Σ is reverse range restricted if each clause in Σ is reverse range restricted.

Examples

The following clauses are all reverse range restricted

$p(x) \rightarrow q(x)$
$\rightarrow q(c)$
$\rightarrow q(x)$
$p(c,x) \wedge p(d,y) \rightarrow q(x,y,z)$

None of the following clauses are reverse range restricted.

$p(x) \rightarrow q(c)$
$p(x) \rightarrow q(y)$
$p(c,x) \wedge p(c,y) \rightarrow q(c,y,z)$

Notation

The name "Reverse Range Restricted" is derived from Manthey and Bry's (1988) use of the term "Range Restricted" with respect to their theorem prover "SATCHMO". Their definition of "Range Restricted" is exactly the reverse of my definition: $\mathbf{C}^i$ *is range restricted if every variable that appears in* H^i *appears in some* B^i_j.

That reverse range restriction implies alphabetical monotonicity can be seen very easily, although I omit a formal proof in this section, giving it instead as Corollary 6.2.4.4. Consider any UR-resolution proof that $\mathbf{R}, \mathbf{R}' \models_\Sigma \rho$. If any restriction from $\mathbf{R}'$ is used by any clause from Σ, then it contains a name not in ρ. By

reverse range restriction, that name appears in the conclusion of the clause. But then if that conclusion is used in another clause, the conclusion of that clause contains a name not in ρ. And so on, meaning that every conclusion derived from an element of **R′** contains a name not in ρ. So ρ must be derived without the use of **R′**, so $\mathbf{R} \models_{\Sigma} \rho$.

Section 6.2.3.1: Modal Logic Theories Satisfying Reverse Range Restriction

Of the clauses that are used in propositional modal theories, reverse range restriction is satisfied by seriality, reflexivity, and symmetry. However, transitivity does not because in the clause

$$(w\mathcal{R}x \wedge x\mathcal{R}y) \rightarrow w\mathcal{R}y,$$

x appears in the body but not in the head of the clause.

As corollaries of the Reverse Range Restriction Theorem we obtain alphabetical monotonicity for the modal logics consist of any combination of seriality, reflexivity, and symmetry. However, since either seriality or reflexivity is enough for non-vacuity, the only interesting theory that we get alphabetical monotonicity for is symmetry alone, corresponding to the modal logic **KB**.

Since the varying domain and constant domain variants of first order modal logics do not introduce any new clauses into the logic, alphabetical monotonicity holds for these variants of the modal logics with theories containing seriality, reflexivity, and symmetry: that is, the modal logics **D**, **T**, **KB**, **DB** and **B**. This holds whether or not we demand that the domain of each world is non-empty.

The clauses for descending domains and for cumulative domains do not satisfy reverse range restriction. However, recall that the cumulative domain and descending domain clauses imply each other given the clause for symmetry, as I mentioned in Section 2.5.2. Furthermore, their combination yields the same set of valid formulas as the constant domain variant. Therefore, reverse range restriction is enough to show alphabetical monotonicity for any of the first order variants of the symmetric modal logics.

Section 6.2.3.2: Order Sorted Logics

Recall from Section 2.6 that the clauses in a restriction theory which arise from order sorted logics are of one of the following three forms.

$$sort_2(x) \rightarrow sort_1(x)$$

$$sort_1(x_1) \wedge sort_2(x_2) \wedge \cdots \wedge sort_n(x_n) \rightarrow sort_0(f(x_1, x_2, \ldots, x_n))$$

$$\rightarrow sort_1(t)$$

The first type of clause is an expression of part of the sort ordering, the second is a sorted function declaration, and the third clause expresses the non-emptiness of a sort (where t is a name-free term built up from Σ-constants and Σ-functions).

All these types of clauses are definite Horn clauses, so any resulting restriction theory will satisfy the least Herbrand model property. Furthermore, all three types of clause are reverse range restricted. In each case every variable in the body of clause appears in the head. Hence alphabetical monotonicity holds for order sorted logics. If we demand that every sort is non-empty then non-vacuity will hold, but alphabetical monotonicity holds whether or not we do this.

Section 6.2.4: The Marking Theorem

Unfortunately reverse range restriction, although very easy to check, is not satisfied by many theories which do in fact satisfy alphabetical monotonicity. However, the idea behind reverse range restriction can be used in a more subtle form. The idea is to keep track of new names in order to show that they would have to arise in ρ, which is impossible. In reverse range restriction this was formalised by demanding that all names appearing in the body of a clause also appeared in the head. To formalise a

similar idea in a more general way I introduce ''marked literals''.

Definition 6.2.4.1 (Marked Literal)

Given a set of Σ atoms **R** and an associated function *new*, I will say that a Σ-literal is **R**-*marked* if it contains an occurrence of a name in *new*(**R**). Similarly, a name is **R**-marked if it is an element of *new*(**R**) and a term is **R**-marked if it contains an **R**-marked name.

Notation

Given a Σ-literal $\rho(t_1,t_2,\ldots t_n)$ *I will sometimes want to indicate which argument or arguments are marked. I shall do this by dressing the appropriate argument with a hat. For instance if the third argument of four were marked, I would write* $\rho(t_1,t_2,\hat{t}_3,t_4)$. *In a modal theory using the accessibility relation I might write* $a \mathcal{R} \hat{b}$ *to indicate that* b *was marked.*

To enable the definition of marked literals to be used efficiently, I assume that, for each Σ-literal, only certain argument positions may be marked. I will call these positions *markable*. Markable positions play a similar role in proofs of alphabetical monotonicity to the role that eligible positions played in the soundness proof of Σ**S**(*1345*) given alphabetical monotonicity. For the proof of the Marking Theorem I will use as an invariant property that if the conclusion of a clause is marked, it is marked in a markable position.

Definition 6.2.4.1 (Markable argument position, obeys marking)

Any eligible argument position in a Σ-predicate must be markable.

Suppose that each Σ-predicate contains a certain number of *markable* argument positions. For a set of Σ-atoms **R** and associated function *new* I will say that an Σ-atom *obeys* **R**-*marking* if either it is not marked or it is marked in a markable argument position.

For example, consider the modal accessibility relation $\mathcal{R}$. Suppose we demand that only the second argument position is markable. If a_1 and a_2 are unmarked and $\hat{b}_1$ and $\hat{b}_2$ are marked, then $a_1 \mathcal{R} a_2$, $a_1 \mathcal{R} \hat{b}_1$, and $\hat{b}_1 \mathcal{R} \hat{b}_2$ all obey marking. On the other hand, $\hat{b}_1 \mathcal{R} a_1$ does not obey marking because the literal is marked, but not in a markable position.

Definition 6.2.4.2 (Obeys marking)

A definite Σ-clause *obeys marking* if for any set of Σ-atoms **R** and associated function *new* (where *new*(ρ) is always in an eligible argument position in ρ), and for any instance of the clause:

if every antecedent literal obeys **R**-marking then the conclusion does; and

if some antecedent literal is **R**-marked then so is the conclusion.

An example of proving that a clause obeys marking may help to explain the concept. Examples of literals obeying marking or not were given above, and simply involved basic observation to check. Checking that a clause obeys marking is rather more complicated, since we must deal with any set of Σ-atoms **R**. As an example (taken from the proof of Theorem 6.2.5.1), consider the clause for transitivity,

$$x \mathcal{R} y \,,\, y \mathcal{R} z \rightarrow x \mathcal{R} z$$

As usual, the second argument of $\mathcal{R}$ is the eligible argument position. Assume that it is also the only markable argument position. Now to show that this clause obeys marking, we must consider any possible instance of it and any possible set **R**. It is enough to consider only instances of the clause in which both antecedent literals obey **R**-marking. Apart from the trivial case in which neither is **R**-marked, this leaves two cases: that the first or second argument is marked. Since we assume they obey marking, they must be marked in the second argument position.

(i) If the first antecedent literal is **R**-marked, the instance of the clause is of the form:
$x\mathcal{R}y\ ,\ y\,\mathcal{R}\,\hat{t}\ \rightarrow\ x\mathcal{R}\hat{t}$
Here, the consequent $x\mathcal{R}\hat{t}$ is both **R**-marked and **R**-marked in a markable position, and hence both parts of the condition for the clause to obey marking are satisfied.

(ii) If the second antecedent literal is **R**-marked, the instance of the clause is of the form:
$x\mathcal{R}\hat{y}\ ,\ \hat{y}\,\mathcal{R}\,t\ \rightarrow\ x\mathcal{R}t$
We assumed that $\hat{y}\,\mathcal{R}\,t$ obeys **R**-marking: hence it is **R**-marked also in its second argument position: that is t is **R**-marked. Again, this means that both parts of the condition for the clause to obey marking are satisfied.

Theorem 6.2.4.3: Marking Theorem

If it is possible to define markable argument positions for every Σ-predicate such that every clause in Σ obeys marking, then Σ satisfies alphabetical monotonicity.

Proof

First, assume that we have defined the markable argument positions so that every clause in Σ-obeys marking.

Given the soundness and completeness of UR-resolution for theories expressed in Horn clauses, it is enough to show that:

> With the same preconditions as the theorem and alphabetical monotonicity: for any n, if there is a UR-resolution proof of length n of $\mathbf{R},\mathbf{R}' \vDash_\Sigma \rho$ then there is a UR-resolution proof of $\mathbf{R} \vDash_\Sigma \rho$.

I prove this by induction on the length of the UR-resolution proof.

Induction Base

If there is a proof of length 0 of $\mathbf{R},\mathbf{R}' \vDash_\Sigma \rho$ then $\rho \in \mathbf{R}$ or $\rho \in \mathbf{R}'$. But any element of $\mathbf{R}'$ is $\mathbf{R}'$-marked, and by the hypotheses of alphabetical monotonicity, ρ is not $\mathbf{R}'$-marked. Hence $\rho \in \mathbf{R}$, so $\mathbf{R} \vDash_\Sigma \rho$.

Induction Step

The induction hypothesis is if there is a UR-resolution proof of length k of $\mathbf{R},\mathbf{R}' \vDash_\Sigma \rho$ then there is a UR-resolution proof of $\mathbf{R} \vDash \rho$. I must prove the same thing for k+1.

Say we have a proof of $\mathbf{R},\mathbf{R}' \vDash_\Sigma \rho$ of length k+1.

If the proof involves no $\mathbf{R}'$-marked literals then, since every literal in $\mathbf{R}'$ is $\mathbf{R}'$-marked, the proof is also a proof of $\mathbf{R} \vDash_\Sigma \rho$, and I am finished.

Otherwise, suppose some $\mathbf{R}'$-marked literal is used in the proof.

Observe that every literal in the proof obeys $\mathbf{R}'$-marking. One can see this by a simple induction argument since:

> every literal in **R** is not $\mathbf{R}'$-marked and hence obeys marking; and
>
> every literal in $\mathbf{R}'$ is trivially $\mathbf{R}'$-marked; and
>
> every Σ-clause obeys marking, so every derived literal obeys marking.

In particular, if we consider the last conclusion of a clause in which a $\mathbf{R}'$-marked literal was used, that conclusion must be $\mathbf{R}'$-marked itself. As this was the last use of a $\mathbf{R}'$-marked literal, the conclusion cannot be used later in the proof. Neither can this conclusion be ρ, the final conclusion of the proof, since ρ is not $\mathbf{R}'$-marked. As the clause's conclusion is neither the final conclusion of the proof nor is used again in the proof, we can omit the clause completely from the proof. Doing this provides a UR-resolution proof of $\mathbf{R},\mathbf{R}' \vDash_\Sigma \rho$ of length k. By the induction hypothesis there is a UR-resolution proof of $\mathbf{R} \vDash_\Sigma \rho$.

□

The sufficiency of reverse range restriction can now be proved as a simple corollary of the Marking Theorem. If Σ satisfies the reverse range restriction then we can make it satisfy the hypothesis of the Marking Theorem simply by making *every* argument position in every Σ-predicate markable. With every argument position markable, any literal obeys marking trivially. So the definition of a clause obeying marking reduces to its second case: that if an antecedent literal is marked then so is the conclusion. This condition is sure to be satisfied if every variable in the body of the clause appears also in the head. Reverse range restriction ensures exactly this.

Corollary 6.2.4.4: Reverse range restriction implies alphabetical monotonicity

If Σ satisfies reverse range restriction then Σ satisfies alphabetical monotonicity.

□

The Marking Theorem has another useful corollary. If Σ satisfies the assumptions of the Marking Theorem, then so does any subtheory of Σ, and so the subtheory satisfies alphabetical monotonicity. This means that we need only prove the Marking Theorem for maximal theories which satisfy it.

Corollary 6.2.4.5: Subtheory Corollary of the Marking Theorem

If a theory Σ_0 contains a subset of the set of clauses in Σ, and Σ satisfies the conditions demanded by the Marking Theorem, then Σ_0 satisfies alphabetical monotonicity.

Proof

We may use the same markable positions for Σ_0 as for Σ. As every clause in Σ obeys marking, then so does every clause in Σ_0. So, by the Marking Theorem, Σ_0 satisfies alphabetical monotonicity.

□

This corollary means that the Marking Theorem can fail on theories which satisfy alphabetical monotonicity. This is because there are sets of clauses which satisfy alphabetical monotonicity, but contain subsets of clauses which fail alphabetical monotonicity. Indeed, there are rather trivial examples where this happens. There are also non-trivial examples. For instance, as I showed in Section 5.1, Σ**Tab**(*1345*) is unsound where Σ is the combination of symmetry and transitivity, so this theory must fail alphabetical monotonicity. On the other hand, if we add reflexivity the theory satisfies non-vacuity and so alphabetical monotonicity.

Section 6.2.5: Examples of Modal Theories Satisfying the Marking Theorem

In this section I consider first order variants of modal theories containing transitivity: the results for propositional modal logics can be derived simply by using the subtheory corollary of the Marking Theorem. I consider first the combination of transitivity and descending domains, then the combination of transitivity and cumulative domains. As for the propositional modal logics, these results are enough for the constant and varying domain variants of first order logics since these do not introduce any clauses into the theory.

Theorem 6.2.5.1: Alphabetical monotonicity for transitivity + descending domains

> *Any combination of the clauses for transitivity and descending domain satisfies alphabetical monotonicity.*

Proof

I will use the Marking Theorem and the Subtheory Corollary.

There are two Σ-predicates: the accessibility relation $\mathcal{R}$ and the domain predicate *in*. The eligible position in $\mathcal{R}$ is the second argument position, while for *in* only the first argument position is eligible. I make these the only markable positions. Schematically, I will indicate the markable positions as:

$$x \mathcal{R} \hat{y}$$

$$\hat{x}\ in\ w$$

The clauses in Σ are

$$x \mathcal{R} y\ ,\ y \mathcal{R} z\ \rightarrow\ x \mathcal{R} z \tag{1}$$

$$x \mathcal{R} y\ ,\ z\ in\ y\ \rightarrow\ z\ in\ x \tag{2}$$

By the Marking Theorem, it is enough to show that each clause (1) and (2) obeys marking.

(1) Assuming that both antecedent literals obey marking, there are two cases to consider: that the first and second antecedent literals respectively are marked in their second position.

(i) $x \mathcal{R} y\ ,\ y \mathcal{R} \hat{t}\ \rightarrow\ x \mathcal{R} \hat{t}$
Then the conclusion is marked in its second position, as required.

(ii) $x \mathcal{R} \hat{y}\ ,\ \hat{y} \mathcal{R} t\ \rightarrow\ x \mathcal{R} t$
But if $\hat{y} \mathcal{R} t$ obeys marking then it must be marked in its second position. So t is marked, and so the conclusion is marked in its second position, as required.

(2) Again there are two cases to consider.

(i) $x \mathcal{R} y\ ,\ \hat{t}\ in\ y\ \rightarrow\ \hat{t}\ in\ x$
Then the conclusion is marked in its first position, as required.

(ii) $x \mathcal{R} \hat{y}\ ,\ t\ in\ \hat{y}\ \rightarrow\ t\ in\ x$
But if $t\ in\ \hat{y}$ obeys marking then it must be marked in its first position. So t is marked, and so the conclusion is marked in its first position, as required.

□

Theorem 6.2.5.2: Alphabetical monotonicity for transitivity + cumulative domains

Any combination of the clauses for transitivity and cumulative domain satisfies alphabetical monotonicity.

Proof

Again by the Marking Theorem and the Subtheory Corollary.

As well as the eligible positions in $\mathcal{R}$ and *in*, I will make the second argument position of *in* markable. Schematically:

$$x \mathcal{R} \hat{y}$$

$$\hat{x}\ in\ w$$

$$x\ in\ \hat{w}$$

The clauses in Σ are

$$x \mathcal{R} y\ ,\ y \mathcal{R} z\ \rightarrow\ x \mathcal{R} z \tag{1}$$

$$x \mathcal{R} y\ ,\ z\ in\ x\ \rightarrow\ z\ in\ y \tag{2}$$

For (1) and (2) I must show that the clause obeys marking. For (1) the proof is as in the previous theorem, as the change to the *in* predicate do not affect that clause. It remains to show that (2) obeys marking.

(2) Assuming that both antecedent literals obey marking, there are now three cases to consider, corresponding to each possible markable position in the antecedent.

(i) $x \mathcal{R} y\ ,\ \hat{t}\ in\ x\ \rightarrow\ \hat{t}\ in\ y$
Then the conclusion is marked in its first position, which is a markable position.

(ii) $x \mathcal{R} \hat{y}$, $t \; in \; x \rightarrow t \; in \; \hat{y}$
Then the conclusion is marked in the second position, which is a markable position.

(iii) $\hat{x} \mathcal{R} y$, $t \; in \; \hat{x} \rightarrow t \; in \; y$
But if $\hat{x} \mathcal{R} y$ obeys marking then it must be marked in its second position. So y is marked, and so the conclusion is marked in its second position, which is a markable position.

□

Section 6.3: Summary

In this chapter I have given proofs that a number of theories satisfy non-vacuity and/or alphabetical monotonicity. (Recall, from Corollary 5.3.5, that the satisfaction of non-vacuity guarantees the satisfaction of alphabetical monotonicity.) Also I have given some sufficient conditions for the satisfaction of these properties: these sufficient conditions are much easier to check than the definitions of the properties given in Chapter 5.

In particular, I showed that non-vacuity holds in sorted logics where each sort is non-empty, and all first order variants of serial and reflexive modal logics where each world has a non-empty domain. Also, alphabetical monotonicity holds for all sorted logics of the kind described in Section 2.6, and for all first order variants of the modal logics **K4** (where transitivity alone holds) and **KB** (where symmetry alone holds).

It is interesting to note that proofs of alphabetical monotonicity tend to be much harder than proofs of non-vacuity. This reflects the fact that non-vacuity is a much simpler concept. Yet non-vacuity is much the more desirable property, because it gives much greater order independence of expansion rule application in an expansion tree. In the next chapter, where matrix characterisations of validity are derived, this difference will be of great importance.

Chapter 7

Matrix Characterisations of Validity

Given the work on expansion systems for Σ-logics in this thesis, we now know how to determine the validity of formulas. Picking one of the systems I have discussed, perhaps **ΣS**(*145*), we search for a closed tree. There is another way of expressing what we are doing. Recalling that each tree has associated with it a set of positions, we could say that we search for a set of positions for which there is closed tree. If we use the expansion system, this is an uninteresting change of definition. However, if we take the new definition seriously, we can look for ways of determining if there is a closed **ΣS**(*145*) tree without actually constructing one. Examining this is what I do in this chapter.

The arguments I use in this Chapter are Wallen's (1986, 1989) arguments for deriving matrix characterisations of validity from tableaux based proof systems. Although the arguments I use are not new, the results I obtain are new, representing a generalisation of Wallen's (1989) matrix characterisations in two directions: I provide matrix characterisations in the context of restricted quantification and so for a wider range of logics than Wallen considered; and I formally derive matrix characterisations from any propositional countermodel system, thereby defining a variety of analogues of Wallen's matrices.

In Chapter 5, we saw that **ΣS**(*145*) may only be used if non-vacuity holds. Where non-vacuity fails but alphabetical monotonicity does, we may use **ΣS**(*1345*). However, for the considerations in this chapter the condition (*3*) causes problems. Therefore I deal in Section 7.1 with theories where non-vacuity holds, while in Section 7.2 I consider theories where non-vacuity fails but alphabetical monotonicity holds.

In Section 7.3 I introduce informally one way one might represent matrices as derived from analytic tableaux. In Section 7.4 I discuss the advantages of matrix characterisations of validity over the expansion systems they are derived from.

Section 7.1: Matrix Characterisations of Validity with Non-vacuity

In deciding if there is a closed **ΣS**(*145*) tree for a given set of positions, there are two issues to deal with. First we have to determine if there is a closed tree, and then if there is a closed tree for that set of positions which is a **ΣS**(*145*) tree. I will show that these two issues can be dealt with separately. That is, given a set of positions, it is enough to determine if there is any **ΣS**(*145*) tree for that set of positions and if the set of positions gives rise to any closed tree. It turns out that these two conditions ensure that there is a closed **ΣS**(*145*) tree for that set of positions.

Until now, I have considered sets of positions to be defined in terms of trees. However, we can entirely reverse this idea and consider trees to be defined in terms of sets of positions. This has one major advantage for proof search. For each set of positions there are many trees. Therefore we reduce the number of entities to consider. I now define what I mean by a tree for a set of positions.

Definition 7.1.1 (Tree for a set of positions)

If **P** is a set of positions then $\mathcal{T}$ is a *tree for* **P** iff **P** is exactly the set of positions arising in $\mathcal{T}$.

I now need a simple definition, akin to the definition of a complete tree, but taking **P** into account.

Definition 7.1.2 (P-complete branch, tree)

A branch in $\mathcal{T}$ is **P**-complete iff every possible application of a expansion rule to it has either already been applied on the branch, or introduces a position not contained in **P**.

A **P**-complete tree is one in which every branch is **P**-complete.

Although many ΣS(*145*) or ΣS(*1345*) trees may be trees for the same set of positions, the structure of these trees is not at all simple. The side conditions (*1*) and (*3*) mean that the order in which expansions occur is important. The structure of all complete trees for a set of positions becomes immensely much simpler if we ignore these ordering issues. Determining closure of a tree can be done independently of determining whether the tree satisfies the side conditions. To this end, I use a proof system without any side conditions.

Given a propositional expansion system **S**, the expansion system Σ**S** is the system containing all the propositional rules of **S** together with the quantifier rules (γ), and (δ). These latter rules are as defined in Chapter 4, but with *no side conditions*. Not only does Σ**S** have exactly the same expansion rules as ΣS(*145*) and ΣS(*1345*), but whether or not a tree is closed does not depend on the order of application of the expansion rules. The following theorem demonstrates this.

Theorem 7.1.3: Order Independence Theorem for ΣS

For a given set of positions **P**, *if there is a closed* Σ**S** *tree for* **P**, *then every* **P**-*complete* Σ**S** *tree for* **P** *is closed.*

Proof

Suppose that $\mathcal{T}^1$ is any **P**-complete tree for **P** and that $\mathcal{T}^2$ is a closed Σ**S** tree for **P**. It is enough to show that any complete branch in $\mathcal{T}^1$ contains a superset of the formulas on some branch in $\mathcal{T}^2$.

Picking a particular complete branch containing formulas U in $\mathcal{T}^1$, I work by induction through the structure of the tree $\mathcal{T}^2$. The base is easy, since the root of $\mathcal{T}^1$ and $\mathcal{T}^2$ is the same and the root belongs to every branch of every tree.

Now suppose that the set of formulas V on a branch to some node in $\mathcal{T}^2$ is a subset of U. If the branch is **P**-complete I have finished. If the branch is not complete, then some expansion rule must be applied. But since $V \subseteq U$, the same expansion could be applied to U. As U is the set of formulas on a **P**-complete branch, the expansion must have been applied on the branch. Therefore the new formulas on at least one of the new branches are already contained in U.

□

The Order Independence Theorem has the crucial consequence that to determine the the existence of any closed tree we need only check an arbitrary **P**-complete Σ**S** tree. Indeed, with the definition below, we need merely check that each "path" is closed. Of course this definition is not suitable for direct implementation. For more sensible discussions of paths see Bibel (1987) or Wallen (1989).

Definition 7.1.4 (path)

The set of formulas on a **P**-complete branch in any Σ**S** tree for **P** is called a **P**-*path*.

The title of the chapter refers to a "matrix characterisation of validity", a phrase borrowed from Wallen (1989). A matrix is now easy to define: however I shall not consider suitable representations for matrices or suitable algorithms for dealing with them.

Definition 7.1.5 (matrix)

A matrix for a set of formulas is (some suitable representation for) a set of positions **P** for that set together with the set of all **P**-paths.

The problem of finding a closed tree has been reduced to that of checking all paths in a matrix for closure. For much more on representing matrices and paths, and testing all paths for closure, see Bibel (1987) or Wallen (1989). But we still have to determine whether or not there is some Σ**S** tree which is also a tree in a sound proof system. That is: given **P**, is there a tree for **P** which satisfies (*1*), (*4*), (*5*), and, if appropriate, (*3*)?

Now the principal reason for my introduction of the side condition (*5*) is apparent. Unlike the highly order dependent condition (*2*), the order of expansions in a tree does not affect (*5*) at all. To check a γ-expansion introducing $\gamma_0(t)$, we must check that $\gamma_r(t)$ is LHM-present. But being LHM-present is defined only in terms of **P**. Thanks to the work in Chapter 5, that was easy! Similarly, (*4*) is defined solely in terms of **P**.

All that remains is checking the satisfaction of (*1*) and (*3*). For the rest of this section, using arguments due to Wallen (1986, 1989), I show how we can deal with this. I will make two simple observations about when one position must appear after another on any branch in which they both appear, if (*1*) or (*3*) is to be satisfied. I will then define a binary relation $\lhd$ in terms of **P**, and show that if there is some tree for **P** satisfying (*1*) then $\lhd$ must be irreflexive. It will turn out that in fact the irreflexivity of $\lhd$ is also a *sufficient* condition for (*1*) to be satisfied in some **P**-complete tree. If we know that the set of complete **P**-paths is closed, we know that the **P**-complete tree is closed, from Theorem 7.1.3.

In fact there is a subtlety in the argument I will give. The intuitive justification for the irreflexivity of $\lhd$ does not quite work in Σ**S**(*145*). I need the extra condition (*3*). So in fact I will show that $\lhd$ is irreflexive if there is a Σ**S**(*1345*) tree for **P**. On the other hand, I will show that if $\lhd$ is irreflexive then there is a **P**-complete Σ**S**(*145*) tree, although perhaps not a Σ**S**(*1345*) tree. This subtlety is precisely the reason why I defined non-vacuity. Recall that if Σ satisfies this condition, Σ**S**(*145*) and Σ**S**(*1345*) are equivalent, and the two proofs I give are sufficient. Unfortunately, there are important theories, such as the empty theory, which do not satisfy non-vacuity. For those, a more complicated approach is necessary, which I leave until the next Section (7.2).

Notation

> *Given two positions p_1 and p_2, the correct informal reading of "$p_1 \lhd p_2$" is "for any occurrence of p_2 in a tree for* **P***, p_1 must have appeared earlier on the same branch". Less long windedly, one might say "p_2 must appear after p_1", although perhaps keeping in mind that this loses some nuances of the longer version.*

It is convenient to define two binary relations to express the information each observation gives us about the ordering of expansions, and a third to combine the first two.

The first observation is that no position can appear on a branch if it is a subposition of a formula that has not yet appeared on a branch. I have already defined a binary relation to express this, in Definition 3.5.2. For convenience I recall the salient part of the definition.

Definition 3.5.2 ($\ll$)

> If p_2 is a subposition of p_1, then we write $p_1 \ll p_2$.

The second observation applies if two positions both introduce the same name and one is a δ_0-type position $\delta_0(a)$. If so, then the other cannot also be a δ_0-type position if (*4*) is satisfied, and so it must be $\gamma_0(t)$. But certainly if t contains a, then $\delta_0(a)$ position must come before $\gamma_0(t)$ on any branch if we are to satisfy (*1*). This is formalised by the binary relation $\sqsubset$.

Definition 7.1.6 ($\sqsubset$)

> Given a set of positions **P**, $\sqsubset_{\mathbf{P}}$ (or just $\sqsubset$ if **P** is understood) is the smallest binary relation on **P** satisfying:
>
> > if a δ_0-type position $\delta_0(a)$ and a γ_0-type position $\gamma_0(t)$ both belong to **P**, and a appears anywhere in t, then $\delta_0(a) \sqsubset_{\mathbf{P}} \gamma_0(t)$.

Having defined two binary relations on **P** that tell us about the order of expansions, we can combine the information they contain and call the resulting binary relation "$\lhd$".

Definition 7.1.7 ($\lhd$)

$\lhd$ is a binary relation on **P** that is the transitive (but not reflexive) closure of $\ll$ and $\sqsubset$. Symbolically,

$$\lhd =_{def} (\ll \cup \sqsubset)^{+}$$

It is easy to show that $\lhd$ must be irreflexive. Remember the intuitive reading of "$p \lhd q$": that q must appear after p. But if $p \lhd p$ then p must appear after itself, which is absurd. This is the basis for the following lemma.

Lemma 7.1.8

If $\mathcal{T}$ is a Σ**S**(*1345*) tree for a set of positions **P**, then $\lhd$ is irreflexive.

Proof

First note that if $p \ll q$ then p appears on every branch that q appears on. The same holds for $p \sqsubset q$ because if it did not then the γ-expansion introducing q would break (*3*). Also, on every branch on which they do appear p must appear higher, by the observations which motivated the definitions of $\ll$ and $\sqsubset$.

Now suppose, for a contradiction, that for some p_1, we had $p_1 \lhd p_1$. Then there would be positions $p_1, p_2, \ldots, p_k, p_{k+1} = p_1$, such that for each i, either $p_i \ll p_{i+1}$ or $p_i \sqsubset p_{i+1}$. Consider any branch on which, say, p_{k+1} appears. Then, from the first paragraph, p_k must appear higher on the branch. Similarly, $p_{k-1}, \ldots, p_2$ must appear higher on the branch than p_{k+1}. Indeed p_1 must appear higher on the branch than p_{k+1}. Since $p_1 = p_{k+1}$, this is impossible, and we have a contradiction. The supposition that $\lhd$ is not irreflexive is impossible.

□

I will now show how, if $\lhd$ is irreflexive, to construct a **P**-complete tree satisfying (*1*). This tree will not necessarily satisfy (*3*), even though I assumed that (*3*) was satisfied for the previous lemma. The tree is constructed by always introducing $\lhd$-minimal positions. With this result, and the previous one, the importance of the irreflexivity of $\lhd$ becomes clear.

Definition 7.1.9 ($\lhd$**-minimal, satisfies** $\lhd$)

A position p is $\lhd$-minimal in a set of positions if there is no q in that set such that $q \lhd p$.

At any point in a Σ**S** tree, an expansion *satisfies* $\lhd$ if and only if any positions it introduces are $\lhd$-minimal in the set of positions that may be introduced at that point. (The definitions of $\lhd$ and the expansion rules ensure that if one position a expansion introduces is minimal then all are.)

Lemma 7.1.10

If a set of positions **P** for a set of formulas S is such that $\lhd$ is irreflexive, then there is a **P**-complete Σ**S**-tree for S in which each expansion satisfies $\lhd$.

Proof

We construct a Σ**S**-tree by starting with S. At each node, certain expansions only introduce positions in **P** and positions that have not yet appeared on the branch. Say that U is the set of positions which can be introduced by these expansions. If in fact U is empty, then the branch is **P**-complete. Otherwise, we choose an expansion which only introduces positions which are $\lhd$-minimal in U. If there is choice of expansions, any arbitrary choice will do. To show that any resulting tree is **P**-complete, it is enough to consider a node for which U is non empty, and to show that the method allows at least one expansion to be applied. But since U is finite and $\lhd$ is irreflexive, there must be some p in U which is $\lhd$-minimal in U.

□

Lemma 7.1.8 showed that the irreflexivity of $\lhd$ is necessary for the existence of a complete tree satisfying (*1*). Perhaps surprisingly, it is also sufficient, as the following lemma shows.

Lemma 7.1.11

If $\mathcal{T}$ is a **P**-complete Σ**S** tree satisfying $\lhd$, then every δ-expansion in $\mathcal{T}$ satisfies (*1*).

Proof

Consider any δ-expansion introducing $\delta_0(a)$. It is enough to show that no γ-expansion earlier on the same branch can introduce a term t containing a. Suppose, for a contradiction, that $\gamma_0(t)$ were $\lhd$-minimal in the set U of formulas that could have been introduced, but that t contains a. Now $\delta_0(a) \sqsubset \gamma_0(t)$ and so $\delta_0(a) \lhd \gamma_0(t)$. For $\gamma_0(t)$ to be $\lhd$-minimal in U, $\delta_0(a)$ cannot belong to U. But then $\delta_0(a)$ must be a subposition of some position p in U, as otherwise it could not appear on the branch. Then $p \ll \delta_0(a)$ and $\delta_0(a) \sqsubset \gamma_0(t)$ and by transitivity, $p \lhd \gamma_0(t)$. This contradicts the supposition, which was what was wanted.

□

Where non-vacuity holds, the preceding arguments can be put together rather simply, given that (*4*) and (*5*) can be checked globally. The result is a generalisation of Wallen's matrix characterisations of validity for modal logics, excepting the logics **K** and **K4** where non-vacuity fails.

Theorem 7.1.12: Matrix Theorem with Non-vacuity

If Σ satisfies the least Herbrand model property and non-vacuity, and S is a finite set of formulas, then S is Σ-unsatisfiable if and only if there is a matrix for S with a set of positions **P** *for S satisfying:*

(i) *each* **P**-*path is closed; and*

(ii) *there is a one to one function* $f : \mathbf{P}^{\delta} \rightarrow$ *Names such that if* $\delta_0(a) \in \mathbf{P}$, $f(\delta) = a$; *and*

(iii) $\lhd$ *is irreflexive; and*

(iv) *for each* $\gamma_0(t) \in \mathbf{P}$, $\gamma_r(t) \in LHM_{\Sigma}(\mathbf{P})$.

Proof

(If)

Suppose that the conditions are satisfied. By (iii) and Lemmas 7.1.10 and 7.1.11 there is a **P**-complete Σ**S** tree $\mathcal{T}$ satisfying (*1*). From (ii) and (iv), $\mathcal{T}$ also satisfies (*5*) and (*4*). Since $\mathcal{T}$ is **P**-complete, the set of formulas on each branch is a **P**-path. By (i) all these are closed, so each branch in $\mathcal{T}$ must be closed. Therefore $\mathcal{T}$ is a closed Σ**S**(*145*) tree.

At this point it is necessary to use the soundness of Σ**S**(*145*) given non-vacuity, as shown in Theorem 5.3.7, enabling us to deduce that S is Σ-unsatisfiable. There is another condition on Theorem 5.3.7, however, which is that the set of restrictions in $\mathcal{T}$ (that is, the set $\mathbf{P}^{\delta_r}$) must be name-acyclic (Definition 5.2.1). Fortunately, the irreflexivity of $\lhd$ ensures this: we can order $\mathbf{P}^{\delta_r}$ in any way that respects the partial order defined by $\lhd$. For suppose, for a contradiction, that in such an ordering $\delta_r^1(a_1)$ appeared before $\delta_r^2(a_2)$ but that the name a_2 appeared anywhere in $\delta_r^1(a_1)$. Then a_2 would have been introduced by some γ-expansion introducing γ_0 of which δ_r^1 was a subposition. That is, $\gamma_0 \ll \delta_r^1$. Also, by definition, $\delta_r^2 \sqsubset \gamma_0$. But then $\delta_r^2 \lhd \delta_r^1$, which contradicts our supposition. So indeed $\mathbf{P}^{\delta_r}$ is name-acyclic and so Theorem 5.3.7 can be applied.

(Only if)

By the completeness of Σ**S**(*1345*) given non-vacuity (Theorem 5.3.1), there is a closed Σ**S**(*1345*) tree for S, associated with the set of positions **P**. From the Order Independence Theorem (7.1.3), each **P**-path is closed. This gives us (i). Since $\mathcal{T}$ is a Σ**S**(*1345*) tree, (ii) and (iv) are satisfied. From Lemma 7.1.8, $\lhd$ is irreflexive, giving us (iii).

□

Section 7.2: Matrices where Non-vacuity Fails

The proof of the Matrix Theorem with non-vacuity relies on the soundness of $\Sigma\mathbf{S}(145)$. This only holds if non-vacuity does. The essential feature of the proof systems that makes matrices a sensible idea is the use of the side condition (5) since it does not rely on the position of an expansion to determine its satisfaction. Without non-vacuity, the only proof system containing this side condition is $\Sigma\mathbf{S}(1345)$. Since important logics fail non-vacuity, such as the modal logic **K** and sorted logics with possibly empty sorts, I now turn to deriving a Matrix Theorem for these logics. The difference with the previous case is that we must worry about satisfying (3) given a set of positions.

The problem with paths as I defined them before is that positions such as $\gamma_0(t)$ can arise on a path even if t contains names that are not on the path. This means that no $\Sigma\mathbf{S}(1345)$ tree could be built containing a branch for that path, since (3) would never allow $\gamma_0(t)$ to be introduced. The solution is in fact extremely simple. We just ignore such positions when testing the closure of a path. We could redefine paths to take account of (3). However this would reintroduce some order dependence in the construction of paths, going against the spirit of matrices. Instead it is sufficient to build paths as before, but ignore formulas in paths that could not satisfy (3).

I must derive a syntactic test for a path corresponding to a closed branch when (3) applies. Fortunately, if a set of positions is not expandable, it is easy to show that all $\lhd$-minimal positions that could be introduced by a $\Sigma\mathbf{S}$ rule have a certain syntactic property. I show this in Lemma 7.2.1. When $\lhd$ is irreflexive, this is enough because in a set of positions, any position is $\lhd$-greater than some $\lhd$-minimal position. So the syntactic check becomes checking that every position involved in closing a path is not $\lhd$-greater than some position which has the property that ensures it could not be introduced. All this should become clearer as the details are filled out below.

Lemma 7.2.1

Suppose that V is the set of positions on a **P**-complete branch in a $\Sigma\mathbf{S}(3)$ tree, and that U is the set of positions on a complete $\Sigma\mathbf{S}$ branch extending V. Then each $\lhd$-minimal position in $U - V$ is of the form $\gamma_0(t)$, where t contains a name a that is not introduced into a δ_0 position $\delta_0(a)$ in U.

Proof

First, any $\lhd$-minimal position in $U - V$ could be introduced by a $\Sigma\mathbf{S}$ expansion since it would not be a subposition of any other position in $U - V$. Hence if it cannot be introduced into V it is because it arises from a γ-expansion breaking (3). So any $\lhd$-minimal position in $U - V$ is $\gamma_0(t)$ and contains a name a that is not in V. But the name a cannot appear in $\delta_0(a) \in U - V$, as otherwise by definition $\delta_0(a) \sqsubset \gamma_0(t)$ making $\gamma_0(t)$ non $\lhd$-minimal.

□

When $\lhd$ is irreflexive we derive a test for whether or not a position in a path can appear in a corresponding branch in a $\Sigma\mathbf{S}(1345)$ tree. After all, if $\lhd$ is irreflexive then every position has a $\lhd$-minimal position that is $\lhd$-less than it. The following definition captures this idea.

Definition 7.2.2 (satisfying (3) in a path, (3)-closed path)

If U is a **P**-path for S, and p is a position in U, then p *satisfies* (3) *in* U iff for each γ_0-position $\gamma_0(t)$ in **P** such that $\gamma_0(t) = p$ or $\gamma_0(t) \lhd p$, every name a in t is either ε or appears in some δ_0-position $\delta_0(a) \in U$.

If U is a **P**-path for S, then U is (3)-*closed* iff it contains two complementary positions p and q which satisfy (3) in U.

This is all that is needed to provide a matrix characterisation of validity if non vacuity fails but alphabetical monotonicity does not. Everything else is as it was in Section 7.1. This characterisation of validity applies to the modal logics **K** and **K4**, logics which Wallen (1989) dealt with but which were not

captured in Section 7.1.

Theorem 7.2.3: Matrix Theorem without Non-vacuity

If Σ satisfies the least Herbrand model property and the alphabetical monotonicity, and S is a finite set of formulas, then S is Σ-unsatisfiable if and only if there is a matrix for S containing a set of positions **P** *for S satisfying:*

(i) *each* **P**-*path is (3)-closed; and*

(ii) *there is a one to one function $f : \mathbf{P}^{\delta} \rightarrow$ Names such that if $\delta_0(a) \in \mathbf{P}$, $f(\delta) = a$; and*

(iii) *$\lhd$ is irreflexive; and*

(iv) *for each $\gamma_0(t) \in \mathbf{P}$, $\gamma_r(t) \in LHM_{\Sigma}(\mathbf{P})$.*

Proof

(If)

We construct any **P**-complete Σ**S**(*3*) tree $\mathcal{T}$. From (ii) and (iv) $\mathcal{T}$ satisfies (*5*) and (*4*). Extend $\mathcal{T}$ to a **P**-complete Σ**S** tree $\mathcal{T}'$. Consider any complete branch V in $\mathcal{T}$ and any complete extension U of it in $\mathcal{T}'$. Lemma 7.2.1 implies that any $\lhd$-minimal position in $U-V$ does not satisfy (*3*). As $\lhd$ is irreflexive from (iii), for any position in $U-V$, some $\lhd$-minimal position is $\lhd$-less than it. So no position in $U-V$ satisfies (*3*). But U is (*3*)-closed from (i) and so contains two complementary positions that satisfy (*3*). These two positions must be in V, and hence V is closed. Therefore $\mathcal{T}$ is a closed Σ**S**(*1345*) tree. As in the proof of Theorem 7.1.12, the set of restrictions in $\mathcal{T}$ is name-acyclic. Since Σ satisfies the least Herbrand model property and alphabetical monotonicity, Σ**S**(*1345*) is sound (Theorem 5.2.4). Therefore S is Σ-unsatisfiable.

(Only if)

By the completeness of Σ**S**(*1345*) given alphabetical monotonicity, there is a closed Σ**S**(*1345*) tree for S, associated with the set of positions **P**. Any branch V in this tree is closed. Consider any path U containing V. Since each γ-expansion satisfies (*3*) every position $\gamma_0(t)$ in V satisfies (*3*) in V, and since $V \subseteq U$, in U. Thus we get (i). We obtain (ii), (iii) and (iv) as in the Matrix Theorem with non-vacuity.

□

Section 7.3: An Example: Matrices Based on Tableaux

I have avoided giving any explicit representation for a matrix. In this section I give an example of how one might represent a matrix, at least on paper. I have chosen **Tab** as the starting point because the resulting matrix characterisation of validity is very close to Bibel's (1987) connection method. I will not present proofs that my definitions are correct. If you are interested in representing matrices and checking that all paths are closed, you should turn to Wallen (1989) or Bibel (1987).

First I define one way of constructing matrices.

Propositional Matrix Construction Rules

If a formula X is a literal then X is the matrix for itself.

If M_1 is the matrix for α_1 and M_2 is the matrix for α_2 then $\begin{bmatrix} M_1 & M_2 \end{bmatrix}$ is the matrix for α.

If M_1 is the matrix for β_1 and M_2 is the matrix for β_2 then $\begin{bmatrix} M_1 \\ M_2 \end{bmatrix}$ is the matrix for β.

Definition 7.3.1 (Paths through a Matrix)

$\{X\}$ is the unique path through the literal X.

If π_1 is a path through M_1 and π_2 is a path through M_2 then $\pi_1 \cup \pi_2$ is a path through $\begin{bmatrix} M_1 & M_2 \end{bmatrix}$.

If π_1 is a path through M_1 then π_1 is also a path through $\begin{bmatrix} M_1 \\ M_2 \end{bmatrix}$.

If π_2 is a path through M_2 then π_2 is also a path through $\begin{bmatrix} M_1 \\ M_2 \end{bmatrix}$.

Example (Running example from Chapter 3)

A propositional matrix, for a formula without equivalences, is no more than a different syntax for the formula itself. For example, recall the running example from Chapter 3.

$$\mathbf{1}(P \vee Q) \wedge (P \vee \neg Q) \wedge (\neg P \vee Q) \wedge (\neg P \vee \neg Q)$$

A matrix for this is:

$$\begin{bmatrix} \begin{bmatrix} \mathbf{1}P \\ \mathbf{1}Q \end{bmatrix} & \begin{bmatrix} \mathbf{1}P \\ \mathbf{0}Q \end{bmatrix} & \begin{bmatrix} \mathbf{0}P \\ \mathbf{1}Q \end{bmatrix} & \begin{bmatrix} \mathbf{0}P \\ \mathbf{0}Q \end{bmatrix} \end{bmatrix}$$

There are sixteen paths through this matrix. For example the path picking out the top element in each submatrix is

$$\{\mathbf{1}P\} \cup \{\mathbf{1}P\} \cup \{\mathbf{0}P\} \cup \{\mathbf{0}P\} \;=\; \{\mathbf{1}P, \mathbf{0}P\}$$

while the path picking out the bottom element in the first and fourth columns and the top element in the second and third columns is

$$\{\mathbf{1}Q\} \cup \{\mathbf{1}P\} \cup \{\mathbf{0}P\} \cup \{\mathbf{0}Q\} \;=\; \{\mathbf{1}Q, \mathbf{1}P, \mathbf{0}P, \mathbf{0}Q\}$$

In this example of course all the paths are closed.

End of Example

I should emphasise that the order in which submatrices appear in either a column or row is unimportant.[1] This follows from the order independence theorem (Theorem 7.1.3). One can think of a matrix as a skeleton tableau. Each expansion has been applied, removing the unexpanded formulas to leave only the expanded formulas. All that remains to make a tableau is to choose an order to apply the expansions in. This can be seen by a simple comparison of the tableau rules and matrix construction rules.

Tableau Expansion Rule	Matrix Construction Rule
$\dfrac{\alpha}{\alpha_1 \quad \alpha_2}$	$M(\alpha) \;\Rightarrow\; \begin{bmatrix} M(\alpha_1) & M(\alpha_2) \end{bmatrix}$
$\dfrac{\beta}{\beta_1 \quad \beta_2}$	$M(\beta) \;\Rightarrow\; \begin{vmatrix} M(\beta_1) \\ M(\beta_2) \end{vmatrix}$

The matrix construction rules are just the tableau rules turned through 90°!

Matrices for restricted quantification are more complex. Where propositional matrices in a sense merely rewrite the formula, a first order matrix does more. In a tableau, the (γ) rule may apply many times. A matrix records this information, as well as telling us which terms are introduced on each occasion. The central feature of a matrix is retained: it still expresses the order independence of expansions.

[1] I thank John Derrick for pointing out that this completely breaks the analogy with mathematical matrices (personal communication).

Now it is important to recall that a matrix is defined with a set of positions **P**, and that **P** contains (by definition) the information about how each γ-position is expanded.

Matrix Construction Rules for Restricted Quantifiers

If a quantifier position γ or δ has no subposition in **P** then it is the matrix for itself.

If a δ-type position δ has a subposition $\delta_0(a)$ in **P** then the matrix for $\delta_0(a)$ is the matrix for δ.

If a γ-type position γ has subpositions $\gamma_0(t_1), \ldots, \gamma_0(t_n)$ in **P**, and M_i is the matrix for $\gamma_0(t_i)$ then $\begin{bmatrix} M_1 & \cdots & M_n \end{bmatrix}$ is the matrix for γ.

The definition of path need not be changed.

Note that the restrictions γ_r and δ_r do not appear in a matrix. Even more clearly now, reasoning in the restriction theory has been separated from logical reasoning. At least where non-vacuity holds, one can calculate the least Herbrand model for a set of positions **P** and whether each γ-expansion represents an expansion satisfying (5) without reference to the matrix.

Example

Consider the formula

$$\forall x\ (P(x) \wedge Q(x)) \supset ((\ \forall y\ P(y)) \wedge (\forall z\ Q(z)))$$

This can be proved valid by the following simple matrix for the formula with polarity **0**.

$$\begin{bmatrix} \mathbf{1}P(a) & \mathbf{1}Q(a) & \mathbf{1}P(b) & \mathbf{1}Q(b) & \begin{bmatrix} \mathbf{0}P(a) \\ \mathbf{0}Q(b) \end{bmatrix} \end{bmatrix}$$

The two paths through the matrix are

$$\{\mathbf{1}P(a), \mathbf{1}Q(a), \mathbf{1}P(b), \mathbf{1}Q(b), \mathbf{0}P(a)\}$$

$$\{\mathbf{1}P(a), \mathbf{1}Q(a), \mathbf{1}P(b), \mathbf{1}Q(b), \mathbf{0}Q(b)\}$$

End of Example

Matrices in the first order case contain information both about how many times a γ-type position is expanded, and which terms are introduced each time. The second of these is somewhat out of place: the choice of which terms are introduced should be made by a process of unification. This could be achieved, as Wallen does, by introducing variables instead of terms at universal instantiation and demanding that a suitable assignment of terms to variables be found. However, the first is fundamental: the difficulty of deciding the multiplicity (in Wallen's terminology) of a γ-position is precisely why the first order connection method is undecidable.

Section 7.4: The Reasons for Studying Matrices

In the development of this chapter, I have not discussed why the development is worth studying in the first place. The treatment I have given makes the relationship with automated theorem proving difficult to see, for instance my definition of matrices as "some suitable representation for" the set of all complete paths. The reasons for studying matrices in the way I have done in this chapter are simple: Wallen (1986, 1989) has shown how matrices derived in this way from analytic tableaux lead to analogues of Bibel's connection method, and Bibel (1982, 1987) has shown that his connection method is of great interest to automated theorem proving. The application of matrix characterisations of validity to automated theorem proving was first discussed by Prawitz (1960, 1969).

As well as the link with the connection method, Wallen has given three reasons why matrix characterisations of validity should be more computationally sensitive than the initial tableaux characterisation. These are that the matrix characterisations remove *notational redundancy*, that they reduce the *order dependence* between rule applications, and that they provide naturally for *connection*

rather than connective based search.

The first advantage of a matrix characterisation of validity is that they eliminate some notational redundancy in tableaux. This was achieved by the use of *positions*, introduced in Section 3.5, which removed the need to represent separately different occurrences of the same formula. This gives a major saving of space in a computer implementation. In this thesis I have used a slightly different notion of position to Wallen to simplify my proofs. Wallen's more representational notion of position is partly inherited from Boyer & Moore's (1972) use of "structure-sharing".

A second advantage is that the problem of order dependence between expansion rules is eliminated. I have emphasised this point throughout Chapters 5 and 7. One way of looking at it is to point out that we can apply rules in some canonical order to produce a possibly *unsound* derivation. Then we can apply special tests to see if an equivalent sound derivation exists. Implementing these tests is considerably less expensive than the alternative in the tableau proof system, which is to search through many possible derivations looking for one in which the ordering of expansions comes out right. This is particularly helpful compared to some proof systems for modal logics, such as those described by Fitting (1983 Chapter Two, 1988) in which application of certain rules represents a choice which may be wrong: if it is wrong then a lot of work is wasted.

The third advantage is that, given a matrix, to check for it being closed we have to check each path for closure. This means that search can be directed by looking explicitly for pairs **0**ϕ and **1**ϕ, that is by looking for *connections*. This is a notable improvement on the situation in the tableau system, where at any point we have available for search only the top *connectives* in that sequent, giving us no information on which formula should be expanded next. This emphasis on connection-based rather than connective-based search is the central feature of Andrews' (1981) and Bibel's (1987) work on automated proof search.

The three advantages I have sketched certainly apply to a matrix characterisation of validity based on analytic tableaux. This means that the work in this chapter can be seen as a generalisation of Wallen's (1989) matrix characterisations of validity, generalising from particular (modal) restriction theories to arbitrary theories satisfying certain conditions.

The three advantages discussed above apply equally to other expansion systems such as **KE**. Therefore the work in this chapter generalises Wallen's (1989) results in another way: I have shown how to derive analogues of his matrix characterisations of validity starting from other proof systems. This is useful since it is desirable to derive computationally sensitive proof systems based on, for example, **KE**. Since D'Agostino (1990) has shown that **KE** is preferable in some ways to tableaux, we might derive a better proof method than the connection method. There are two problems which limit the immediate usefulness of my results. First, it is non-trivial to derive an automated theorem proving method from matrix characterisations of validity. This is particularly so in the general case since I have not discussed how matrices should be represented, the presentation in Section 7.3 being limited to matrices based on tableaux only. Second, it is natural in a tableaux based system to use propositional expansions on each branch as much as possible, and Wallen's representation of matrices effectively assumes that this is done. However, in other proof systems this is not so. In **KE** for example, one certainly should not apply the splitting rule (PB) as much as possibly, and this applies to a lesser extent if one uses analytic (PB). Since my definition of a matrix demands that any expansion applied on one branch must be applied on all, we cannot avoid this. It is possible that the advantages D'Agostino shows that **KE** has will be lost if we enforce this demand, but this requires further investigation. At least the work in this chapter represents a significant first step in adapting Wallen's methods to proof systems such as **KE**.

Chapter 8

Instantiation in Restricted Quantification

The two expansion rules (γ) and (δ) allow us to form instances of restricted quantifications, given certain conditions. However, they only allow us to form instances in a formula if the formula itself is a quantification. This is why completeness only held for countermodel expansion systems. A countermodel expansion system guarantees to provide us with an explicit quantification which we can then apply (γ) or (δ) to. This formed the crucial part in the completeness proof, enabling the induction step to be extended from all quantifications of a certain depth to all formulas containing quantifiers of that depth. However, other expansion systems, such as resolution, do not allow this.

To extend non-countermodel expansion systems to restricted quantification, we need rules which allow us to instantiate quantifiers as subformulas of a propositional formula. The reason I have delayed discussing this problem is that the conditions under which *sound* expansion rules for instantiation can be introduced are much more restricted than the conditions under which the simpler rules (γ) and (δ) are sound. The new rules are sound only if the restriction theory satisfies non-vacuity.

The new instantiation rules are quite natural, and essentially the same as the old rules. Instead of only applying to quantifications, though, they apply to any formula, replacing any outermost quantification by an instance of it. This means that we are ignoring the propositional structure of the formula. So the instantiation rule can only be sound if, in some sense, the propositional structure and the quantificational structure can be completely separated. The proofs will rely on swapping the order of quantification rules and propositional rules, and showing that this can be done within the proof systems. That is, I show that quantification rules and propositional rules *permute*.

Exactly what it means to permute two rule applications requires careful definition, and varies depending on the type of rules in question. I introduce and define each type of permutation as I need them in the proofs in this chapter.

The transformations I use in Section 8.2 and Section 8.3 have an interesting feature. In Section 8.2 they are only sound if the condition (*3*) is absent. In Section 8.3 they are only sound if (*3*) is present. This is not contradictory, as it may seem at first sight. These transformations play a proof theoretical role: they are used to justify Σ**InstS**(*1345*) and Σ**InstS**(*145*). We are not interested in applying transformations to real expansion trees. Since I showed in Chapter 5 that Σ**S**(*1345*) and Σ**S**(*145*) are equivalent if non-vacuity holds, we may choose freely whether to demand (*3*) or not.

Section 8.1: New Expansion Rules for Instantiation

Notation

Given formulas X and A, I will use the notation X[A] to indicate that A occurs once in particular position in X. This notation is from Bundy (1983). For convenience the notation is applicable if A does not appear at all in X. A may appear elsewhere in X, but X[A] refers only to one occurrence. If I have earlier mentioned X[A], and B is another formula, then X[B] indicates the formula X with the relevant occurrence of A replaced by B.

New Quantifier Expansion Rules

(γ-Inst)	$\dfrac{X[\gamma]}{X[\gamma_0(t)]}$	(δ-Inst)	$\dfrac{X[\delta]}{\begin{array}{c}\delta_r(a)\\ X[\delta_0(a)]\end{array}}$

where γ and δ are outermost quantifications in $X[\gamma]$ and $X[\delta]$ respectively.

Notation

If **S** *is any propositional expansion system, I will use "*Σ**InstS***" to refer to the expansion system containing all the expansion rules of* **S***, as well as (γ-Inst) and (δ-Inst).*

The same side conditions are applicable, but I only wish to consider (*1*), (*3*), (*4*), and (*5*). I repeat these as a reminder.

(*1*) applies to δ-expansions:

a is a new name that has not appeared on the branch up to the point of the δ-expansion.

(*3*) applies to γ-expansions:

t contains only names that have been introduced by δ-expansions on the branch up to the γ-expansion, except possibly the name ε.

(*4*) There is a well defined and one to one function f:$\mathbf{P}^{\delta}\rightarrow$*Names* such that $f(\delta)$ is the name introduced by into any immediate subposition of δ. That is, for each δ-type position $\delta \in \mathbf{P}$, if $\delta_0(a) \in \mathbf{P}$ then $f(\delta) = a$.

(*5*) applies to γ-expansions:

provided $\gamma_r(t) \in LHM_{\Sigma}(\mathcal{T})$

I will show in this chapter that if Σ satisfies the least Herbrand model property and non-vacuity, then Σ**InstS**(*145*) is sound and complete given any sound and complete propositional system **S.** To do this, I pick a particular propositional system, **Tab**, and show in Section 8.2 that Σ**InstTab**(*145*) is sound and complete given the conditions on Σ. In Section 8.3 I show that we can permute the instantiation rules fully with propositional rules in Σ**InstTab**(*145*). In particular, we can move all instantiations to the top of a tree, separating completely the first order and propositional reasoning. The propositional reasoning in such **Tab**-trees may be replaced by reasoning in any other sound and complete propositional expansion system, giving the desired result.

Section 8.2: Soundness and Completeness of ΣInstTab(*145*)

Lemma 8.2.1

If there is a finite closed Σ**InstTab**(*145*) tree for a set of formulas S (with a name-acyclic set of restrictions), then there is a finite closed Σ**Tab**(*145*) tree for S (also with a name-acyclic set of restrictions).

Proof

As a first step I note that if there is a closed Σ**InstTab**(*145*) tree then there is one in which each branch contains complementary formulas which are either both literals or both quantifications. The only case to consider is a branch is closed because of α- and β-type formulas. But it is easy to check from the definition of uniform notation that any α-type formula's complement is a β-type formula, and furthermore that $\alpha_1{}^c = \beta_1$ and $\alpha_2{}^c = \beta_2$. So we may expand any such branch as follows:

$$\begin{array}{c}\alpha\\ \beta\\ \hline \alpha_1\\ \alpha_2\\ \hline \beta_1 \quad \beta_2\end{array}$$

If this expansion is repeated until it is no longer possible, the result is a tree which is closed as desired.

Now suppose we have a closed **ΣInstTab**(*145*) tree $\mathcal{T}_0$, which is not a **ΣTab**(*145*) tree, and in which each branch is closed either by literals or quantifications. It must contain an application of (γ-Inst) or (δ-Inst) which is not an application of (γ) or (δ). Since the instance of γ or δ must be outermost, the (Inst) rule must apply to either an α- or β-type formula. This gives four cases. Depending on the case, apply one of the four transformations I define below to the subtree below the application of the (Inst) rule. The transformations makes the application of (Inst) apply to a structurally simpler formula. After repeating it enough, the application of (Inst) can only apply to quantifications: it is then an application of either (γ) or (δ), which are known to be sound.

Note

> *In giving the transformations below, I will pretend that the γ or δ subformula of α or β occurs in both α_1 and α_2 or β_1 and β_2. This simplifies the presentation, although in fact the quantification may only appear once. I do this because I do not need to be careful about this, except in one case, where δ appears in α, when I must distinguish between the two possibilities.*

The first transformation moves an application of (γ-Inst) downwards if it applies to an α-formula.

$$\begin{array}{c} \alpha[\gamma] \\ \hline \alpha[\gamma_0(t)] \\ \hline \mathcal{T} \end{array} \quad \Rightarrow \quad \begin{array}{c} \alpha[\gamma] \\ \hline \alpha_1[\gamma] \\ \alpha_2[\gamma] \\ \hline \alpha_1[\gamma_0(t)] \\ \hline \alpha_2[\gamma_0(t)] \\ \hline \mathcal{T}' \end{array}$$

where $\mathcal{T}'$ is the same as $\mathcal{T}$ except that any application of the (α) rule to $\alpha[\gamma_0(t)]$ in $\mathcal{T}$ is deleted in $\mathcal{T}'$.

The next transformation moves an application of (γ-Inst) downwards if it applies to a β-formula.

$$\begin{array}{c} \beta[\gamma] \\ \hline \beta[\gamma_0(t)] \\ \hline \mathcal{T} \end{array} \quad \Rightarrow \quad \begin{array}{c} \beta[\gamma] \\ \hline \begin{array}{c} \beta_1[\gamma] \\ \hline \beta_1[\gamma_0(t)] \\ \hline \mathcal{T}^1 \end{array} \quad \begin{array}{c} \beta_2[\gamma] \\ \hline \beta_2[\gamma_0(t)] \\ \hline \mathcal{T}^2 \end{array} \end{array}$$

where $\mathcal{T}^1$ is the same as $\mathcal{T}$ except that any application of the (β) rule to $\beta[\gamma_0(t)]$ introducing $\beta_1[\gamma_0(t)]$ and $\beta_2[\gamma_0(t)]$ in $\mathcal{T}$ is deleted in $\mathcal{T}'$ and the subtree below that point replaced by the subtree below $\beta_1[\gamma_0(t)]$. $\mathcal{T}^2$ is defined similarly except that the subtree below $\beta_2[\gamma_0(t)]$ is used.

The third transformation moves an application of (δ-Inst) downwards if it applies to an α-formula. Because of the side condition (*1*) I must distinguish between the cases where δ appears in α_1 and in α_2.

$$\begin{array}{c} \alpha[\delta] \\ \hline \delta_r(a) \\ \alpha[\delta_0(a)] \\ \hline \mathcal{T} \end{array} \quad \Rightarrow \quad \begin{array}{c} \alpha[\delta] \\ \hline \alpha_1[\delta] \\ \alpha_2 \\ \hline \delta_r(a) \\ \alpha_1[\delta_0(a)] \\ \hline \mathcal{T}' \end{array} \quad \textit{or} \quad \begin{array}{c} \alpha[\delta] \\ \hline \alpha_1 \\ \alpha_2[\delta] \\ \hline \delta_r(a) \\ \alpha_2[\delta_0(a)] \\ \hline \mathcal{T}' \end{array}$$

where we choose the transformation according to whether δ appears in α_1 or α_2;

and where $\mathcal{T}'$ is the same as $\mathcal{T}$ except that any application of the (α) rule to $\alpha[\delta_0(a)]$ in $\mathcal{T}$ is deleted in $\mathcal{T}'$.

The last transformation moves an application of (δ-Inst) downwards if it applies to an β-formula.

$$\begin{array}{c} \beta[\delta] \\ \hline \delta_r(a) \\ \beta[\delta_0(a)] \\ \hline \mathcal{T} \end{array} \quad \Rightarrow \quad \begin{array}{c} \beta[\delta] \\ \hline \begin{array}{c} \beta_1[\delta] \\ \hline \delta_r(a) \\ \beta_1[\delta_0(a)] \\ \hline \mathcal{T}^1 \end{array} \quad \begin{array}{c} \beta_2[\delta] \\ \hline \delta_r(a) \\ \beta_2[\delta_0(a)] \\ \hline \mathcal{T}^2 \end{array} \end{array}$$

where $\mathcal{T}^1$ is the same as $\mathcal{T}$ except that any application of the (β) rule to $\beta[\delta_0(a)]$ introducing $\beta_1[\delta_0(a)]$ and $\beta_2[\delta_0(a)]$ in $\mathcal{T}$ is deleted in $\mathcal{T}'$ and the subtree below that point replaced by the subtree below $\beta_1[\delta_0(a)]$. $\mathcal{T}^2$ is defined similarly except that the subtree below $\beta_2[\delta_0(a)]$ is used.

Note

This transformation would not necessarily be sound if (*3*) *were demanded.* δ *can only appear in one of* β_1 *and* β_2. *This means that the new branch below the other* β_i *no longer contains the* δ*-expansion (recall that I assumed that the transformation above does not indicate that the* δ*-expansion appears on both branches). Then if some* γ*-expansion appears on that branch and introduces the same name a as the* δ*-expansion, the* γ*-expansion would no longer satisfy* (*3*).

In each case it is easy to check that, since the original application of the (Inst) rule satisfies (*1*), (*4*), and (*5*), the new applications will. Therefore the result of the transformation is a new **ΣInstTab**(*145*) tree $\mathcal{T}_1$.

This transformation may be repeated if there are still applications of (Inst) that are not applications of (γ) or (δ), giving trees $\mathcal{T}_1, \mathcal{T}_2, \ldots$ But it may only be repeated finitely often, and when it cannot be repeated we end up with a **ΣTab**(*145*) tree $\mathcal{T}_n$. It only remains to show that $\mathcal{T}_n$ is closed.

The only way that $\mathcal{T}_i$ could be closed and $\mathcal{T}_{i+1}$ not closed is if the closure depends on the formula removed. The only formula that the transformations remove from a branch are, in the four cases, $\alpha[\gamma_0(t)]$, $\beta[\gamma_0(t)]$, $\alpha[\delta_0(a)]$, or $\beta[\delta_0(a)]$. But none of these formulas are literals or quantifications, and we chose $\mathcal{T}_0$ to be closed by literals or quantifications. Therefore each $\mathcal{T}_i$ is closed.

As to the preservation of name-acyclicity: none of the transformations affect the set of restrictions in the tree, so name acyclicity is unaffected.

□

Theorem 8.2.2: Soundness and Completeness of ΣInstTab(*145*) and ΣInstTab(*1345*)

If Σ *satisfies the least Herbrand model property and non-vacuity, and S is a set of* Σ*-formulas, then* Σ *is unsatisfiable if and only if there is a closed* **ΣInstTab**(*145*) *tree for S containing a name-acyclic set of restrictions.*

Proof

(Completeness)

As seen in Chapter 4, given the least Herbrand model property, **ΣTab**(*1345*) is complete. But any **ΣTab**(*1345*) tree is also a **ΣInstTab**(*1345*) tree as well as a **ΣInstTab**(*145*) tree containing a name-acyclic set of restrictions. Therefore these latter two proof systems are also complete.

(Soundness)

From Lemma 8.2.1, if there is a closed **ΣInstTab**(*145*) tree for *S* containing a name-acyclic set of restrictions then there is also a closed **ΣTab**(*145*) tree for *S* containing a name-acyclic set of restrictions. From Theorem 5.3.7, **ΣTab**(*145*) is sound given non-vacuity, so *S* must be unsatisfiable. Any **ΣTab**(*1345*) tree is also a **ΣTab**(*145*) tree so soundness applies to **ΣTab**(*1345*) too.

□

Section 8.3: Soundness and Completeness of ΣInstS(*145*) and ΣInstS(*1345*)

The reason I chose to use a specific expansion system, **Tab**, in Section 8.2 is that I can generalise Theorem 8.2.2 to apply to *any* expansion system **S**. Note that this includes expansion systems which are not countermodel expansion systems, and with which I have been unable to deal so far.

In this section, I demonstrate the soundness of ΣInstS(*145*) in Theorem 8.3.1. In Lemma 8.3.2 I show how to move all applications of the instantiation rules to the top of a proof in ΣInstS(*1345*), and use this in Theorem 8.3.1 to demonstrate the completeness of ΣInstS(*145*) and ΣInstS(*1345*).

As in Section 8.2, that a ΣInstS(*145*) tree contains a name-acyclic set of restrictions is necessary for soundness. However, it is tedious to repeatedly mention this since none of the transformations of trees in this section affect the set of restrictions in a tree at all: I omit further mention of this in the proofs in this section.

Theorem 8.3.1: Soundness of ΣInstS(*145*) and ΣInstS(*1345*)

If **S** *is a sound propositional expansion system, and* Σ *satisfies non-vacuity and the least Herbrand model property, then* ΣInstS(*145*) *and* ΣInstS(*1345*) *are sound.*

Proof

I show that any application of the rules (γ-Inst) and (δ-Inst) is sound in any finite closed ΣInstS(*145*) tree.

Suppose, for a contradiction, that there was an example of an unsound application of one of the (Inst) rules on a branch of a ΣInstS(*145*) tree containing formulas U which are satisfiable, introducing a new formula X such that U,X was unsatisfiable. Also suppose that the set of restrictions appearing in the entire tree was **R**. We can easily construct a ΣInstTab(*145*) tree in which the formulas U appear on one branch, **R** on another and in which all other branches are closed. For example, considering U and **R** as conjunctions of the formulas in them, the following formula works:

$$(U \vee (\mathbf{R} \wedge \mathbf{f}))$$

One branch of the ΣInstTab(*145*) tree closes trivially, as it contains **f**. To the other branch we can do the unsound application of (Inst) giving U,X and then close that branch somehow since U, X is unsatisfiable and ΣInstTab(*145*) is complete. Therefore there is a closed ΣInstTab(*145*) tree for the original formula. This contradicts the facts that ΣInstTab(*145*) is sound and U is satisfiable. Hence there can be no unsound application of either (γ-inst) or (δ-inst).

Thus ΣInstS(*145*) is sound. But any ΣInstS(*1345*) tree is also a ΣInstS(*145*) tree, so ΣInstS(*1345*) is sound too.

□

Lemma 8.3.2

If there is a finite closed ΣInstTab(*1345*) tree for a set of formulas S then there is a finite closed ΣInstTab(*1345*) tree for S in which no propositional rule is applied before an instantiation rule on any branch.

Proof

As in the proof of Lemma 8.2.1, I demonstrate the existence of a tree needed by the theorem by applying suitable transformations to the first tree.

In the proof of Lemma 8.2.1, I mentioned that, given a closed ΣInstTab(*145*) tree, there is another in which each branch is closed by either literals or quantifications. The same applies to ΣInstTab(*1345*), but we can go further, finding a tree in which each branch is closed by literals. If a branch is closed by quantification, then by examining the uniform notation it must be by complementary formulas γ and $\delta = \gamma^c$. But we can apply the following expansions:

$$\begin{array}{c} \gamma \\ \delta \\ \hline \delta_r(a) \\ \delta_0(a) \\ \hline \gamma_0(a) \end{array}$$

Since (*3*) and (*4*) are satisfied throughout the branch, the name a will be new on the branch, satisfying (*1*) and $\gamma_r(a)$ will be the same as $\delta_r(a)$ meaning that (*5*) is satisfied by the γ-expansion. But if $\delta = \gamma^c$ then $\delta_0(a) = \gamma_0(a)^c$. Applying these expansions repeatedly, together with the expansions used in the earlier proof, yields a closed Σ**Tab**(*1345*) tree in which each branch is closed by literals.

Now consider any finite Σ**Tab**(*1345*) tree in which each branch is closed by literals. Considering it as a Σ**InstTab**(*1345*) tree, we can apply one of the following transformations if any application of (Inst) follows a propositional rule.

Notation

> *In the following, "$\mathcal{Q}$" represents either a δ- or a γ-type position, while "$\mathcal{Q}_0(t)$" represents either $\delta_0(t)$ or $\gamma_0(t)$. I will omit occurrences of "$\delta_r(a)$" which would appear with $\mathcal{Q}_0(a)$ if $\mathcal{Q} = \delta$.*

The first two transformations deal with the case that the propositional rule does not affect the quantificational rule. They move an application of ($\mathcal{Q}$-Inst) up through an α or β application.

$$\begin{array}{c} \alpha \\ X[\mathcal{Q}] \\ \hline \alpha_1 \\ \alpha_2 \\ \hline X[\mathcal{Q}_0(t)] \\ \hline \mathcal{T} \end{array} \quad \Rightarrow \quad \begin{array}{c} \alpha \\ X[\mathcal{Q}] \\ \hline X[\mathcal{Q}_0(t)] \\ \hline \alpha_1 \\ \alpha_2 \\ \hline \mathcal{T} \end{array}$$

$$\begin{array}{c} \beta \\ X[\mathcal{Q}] \\ \hline \begin{array}{c} \beta_1 \\ \hline X[\mathcal{Q}_0(t)] \\ \hline \mathcal{T}^1 \end{array} \quad \begin{array}{c} \beta_2 \\ \hline X[\mathcal{Q}_0(t)] \\ \hline \mathcal{T}^2 \end{array} \end{array} \quad \Rightarrow \quad \begin{array}{c} \beta \\ X[\mathcal{Q}] \\ \hline X[\mathcal{Q}_0(t)] \\ \hline \begin{array}{c} \beta_1 \\ \hline \mathcal{T}^1 \end{array} \quad \begin{array}{c} \beta_2 \\ \hline \mathcal{T}^2 \end{array} \end{array}$$

I have shown the expansion of $X[\mathcal{Q}_0(t)]$ as happening immediately after the β-expansion on both branches. However, we apply the transformation even if $X[\mathcal{Q}_0(t)]$ is only introduced immediately on one branch. In that case, we replace $\mathcal{T}^i$ after transformation by $\mathcal{T}^{i\prime}$ in which any expansion of $X[\mathcal{Q}]$ to $X[\mathcal{Q}_0(t)]$ is deleted on the other branch.

The second two transformations cover the possibility that the propositional rule does affect the quantificational rule. Intuitively these transformations are the reverse of those used in the proof of Lemma 8.2.1.

$$\begin{array}{c} \alpha[\mathcal{Q}] \\ \hline \alpha_1[\mathcal{Q}] \\ \alpha_2[\mathcal{Q}] \\ \hline \alpha_1[\mathcal{Q}_0(t)] \\ \hline \alpha_2[\mathcal{Q}_0(t)] \\ \hline \mathcal{T} \end{array} \quad \Rightarrow \quad \begin{array}{c} \alpha[\mathcal{Q}] \\ \hline \alpha[\mathcal{Q}_0(t)] \\ \hline \alpha_1[\mathcal{Q}_0(t)] \\ \alpha_2[\mathcal{Q}_0(t)] \\ \hline \mathcal{T}' \end{array}$$

where $\mathcal{T}'$ is formed from $\mathcal{T}$ by deleting any use of $\alpha_i[\mathcal{Q}]$. That is, we delete any application of any rule to any subposition of $\alpha_i[\mathcal{Q}]$ up to and including an application of ($\mathcal{Q}$-Inst). We replace any application of a propositional rule that we have deleted by an application of the same rule to the analogous subposition of

$\alpha_i[Q_0]$.

$$\begin{array}{c} \beta[Q] \\ \hline \begin{array}{cc} \beta_1[Q] & \beta_2[Q] \\ \hline \beta_1[Q_0(t)] & \beta_2[Q_0(t)] \\ \hline \mathcal{T}^1 & \mathcal{T}^2 \end{array} \end{array} \quad \Rightarrow \quad \begin{array}{c} \beta[Q] \\ \hline \beta[Q_0(t)] \\ \hline \begin{array}{cc} \beta_1[Q_0(t)] & \beta_2[Q_0(t)] \\ \hline \mathcal{T}^{1\prime} & \mathcal{T}^{2\prime} \end{array} \end{array}$$

where $\mathcal{T}^{i\prime}$ is formed from $\mathcal{T}^i$ by removing any use of $\beta_i[Q]$ as in the previous transformation. Also, we delete any expansion of $\beta_i[Q]$ to $\beta_i[Q_0(t)]$ if the expansion is not immediate as shown here (exactly as we did in the earlier transformation through a β expansion in this proof).

All these transformations yield trees which still satisfy the conditions (*1345*). Satisfaction of (*4*) and (*5*) are unaffected, and (*3*) is still satisfied because γ-expansions are only pushed up through propositional rules, thereby not changing the set of names on the branch: (*3*) and (*4*) together ensure that (*1*) is satisfied. Note the essential use of (*3*). Otherwise, permuting (γ-Inst) applications up through (β) applications might lead to trees breaking (*1*), if a name introduced from (γ) clashed with a name introduced by some (δ)-application on the other branch below the β-application.

We can repeat the transformations until they are impossible: this only happens when no propositional rule precedes a quantification rule.

The only formulas which are removed from a branch by any transformation are $\alpha_i[Q]$ or $\beta_i[Q]$, and this only happens if it actually contains Q. But if it contains Q then it is not a literal. So no literals are ever removed from a branch. Since the original tree is closed by literals, so is the final one.

□

Theorem 8.3.3: Completeness of ΣInstS(*145*) and ΣInstS(*1345*)

If **S** *is a complete propositional expansion system, and* Σ *satisfies the least Herbrand model property, then* **ΣInstS**(*145*) *and* **ΣInstS**(*1345*) *are complete.*

Proof

If a set of formulas *S* is unsatisfiable, I must show that there is a closed **ΣInstS**(*1345*) tree for *S*, which will also be a closed **ΣInstS**(*145*) tree. Recall that **ΣTab**(*1345*) is complete given the least Herbrand model property. So there is a finite closed **ΣTab**(*1345*) tree for *S*. By Lemma 8.3.2 we can find one in which all quantification rules precede all propositional ones. The quantification rules do not split a branch, so after their application we have an expanded set S' which can be closed by applications of rules from **Tab**. That is, S' is *propositionally* unsatisfiable. But we assumed that **S** is propositionally complete, and hence we can find a closed **S** tree for S'. Putting the quantifications and the **S** tree together yields a closed **ΣInstS**(*1345*) tree for *S*.

□

Observe that there is no essential use of the fact that **S** is an expansion system in the above proof. Indeed one could easily prove a similar result for any propositional proof procedure, expansion system or not. The point of the work in this chapter is not in fact proof-theoretic, although I have used proof theoretic methods to derive it. The point is that I have effectively defined what *instantiation* is in restricted quantification if non-vacuity holds: it can be seen as the process of applying the rules (γ-Inst) and (δ-Inst).

Section 8.4: A Sketch of Skolemisation

The proof systems containing instantiation presented in this chapter have two features which make them undesirable for direct implementation in automatic theorem proving programs. First, the instantiation rules are only allowed to instantiate an outermost quantification in a formula. Second, the instantiation rules, as are all quantifier rules in this thesis, are required to introduce explicit names or terms. These two

features mean that any implementation of these proof systems would, at each instantiation, have to *guess* a suitable name or term to be introduced. This is a hopelessly inefficient method of choosing terms. Any method which allows the choice of term to be more *calculated* is a great advantage: all work on resolution based theorem provers, going back to Robinson (1965) takes advantage of this insight.

The central feature of resolution theorem proving methods is the use of skolemisation and unification. It is particularly desirable to consider the application of these techniques at this point because resolution is a non-countermodel expansion system, so the work of Chapters 4 to 7 is not applicable to resolution. Fortunately, skolemisation and unification can be incorporated into ground based systems using standard methods. Reeves (1987) and Fitting (1990b, Chapter 7) discuss the addition of skolemisation and unification to ground based proof systems. In a sentence, the technique is to replace universal instantiation by the introduction of a dummy variable, and existential instantiation by the introduction of a skolem function of the free variables at that point: the choice of term for a universal instantiation can be made at a later stage using unification.

Unfortunately, Reeves and Fitting's presentation cannot be followed directly here. The problem is that the condition (*5*) on universal instantiation limits those terms that may be introduced. Introducing dummy variables instead may lead to adding restrictions via existential introduction that are not justified. For example, suppose one had a formula

$$\forall x_{:\rho_1(x)}\ \exists y_{:\rho_2(x,y)}\ \phi(x,y)$$

It may not be possible to introduce, say, the name a for x if we cannot prove $\rho(a)$. However, if one introduced a dummy variable for x, x' for example, one would have

$$\exists y_{:\rho_2(x',y)}\ \phi(x',y)$$

allowing, with the skolem term $f(x')$ replacing y, the introduction of the restriction

$$\rho_2(x',f(x'))$$

It might then be possible to replace x' with a, and to use the restriction $\rho_2(a,f(a))$ to satisfy the restriction $\rho_1(a)$. This would be unsound. The satisfaction of $\rho_1(x')$ is a precondition for the introduction of $\rho_2(x',f(x'))$: naive skolemisation leads to the possibility of circular reasoning. Frisch (1991, Appendix B) shows how this problem may be avoided. In the above circumstances, instead of introducing the simple restriction $\rho_2(x',f(x'))$, Frisch adds to the restriction theory the clause

$$\rho_1(x') \rightarrow \rho_2(x',f(x'))$$

In the context of this chapter, this could be done equally well. The crucial point that the restriction theory must satisfy non-vacuity would not be affected by such clauses, and all such clauses are definite Horn clauses, and so do not affect the satisfaction of the least Herbrand model property. Therefore one could apply skolemisation and unification to the proof systems in this chapter with little theoretical work. The presence of the restriction theory underlying universal instantiation can in some cases be replaced by the use of special purpose unification algorithms, as discussed by Frisch and Scherl (1990, 1991). A final, minor, point is that the requirement of name-acyclicity as a precondition for soundness would be rendered unnecessary by skolemisation, just as it was rendered unnecessary in Chapter 7 by the use of the binary relation $\lhd$.

If the above sketch were to be filled out, one would arrive at proof systems which are essentially those of Frisch's ''substitutional framework'' (1989, 1991). Indeed, Frisch (1991, Appendix B) explicitly discusses the skolemisation process and the conditions under which it is sound, which are effectively the same as my condition of non-vacuity. So the work in this chapter cannot be seen as a significant new contribution. However, it is still interesting because it approaches the same problem as Frisch from the reverse direction. That is, while Frisch makes skolemisation central and proves its correctness semantically, I consider skolemisation as an essentially syntactic process. With respect to skolemisation, Frisch's work has the considerable advantage over this work that, unlike this sketch, it is complete!

Chapter 9

A More General System for Restricted Quantification

In Chapter 4 I proved that two expansion rules, (γ) and (δ), for dealing with restricted quantification were sound. I also showed that adding them to a propositional countermodel expansion system leads to a complete system, provided that the restriction theory satisfies the least Herbrand model property. This result provided the foundation for all the work in Chapters 5 to 8.

The least Herbrand model property is not always satisfied, of course. In this chapter I prove complete another expansion system given an arbitrary first order restriction theory. The rule (γ) is replaced by a new rule (γ-branch), so called because unlike the old rule, it is a branching rule. Unfortunately, the fact that (γ) is a linear rule is crucial to the close analogy between classical proof systems for first order logic and the proof systems for restricted quantification in this thesis.

There are compelling reasons for the study of the system I introduce in this chapter. There are a number of modal logics whose restriction theory cannot be written in Horn clauses. Examples are counterfactual logics and temporal logics. Sequent and tableau style proof systems for the former have been studied by de Swart (1983) and me (Gent 1990). Temporal logics have been the subject of more intense study because of the natural link between the underlying ideas of time in temporal logics and time in computer systems. Authors who have studied the sequent and tableau style proof systems for temporal logics include Paech (1988) and Goré (1991). Unfortunately, all these proof systems suffer from the computational disadvantages discussed by Wallen (1989) and sketched briefly by me in Section 7.4. A further disadvantage of the mentioned work is that special arguments are required to prove correct the system for each logic. A mistake in one of de Swart's proof systems went unnoticed for six years until I saw it and corrected it (Gent 1990).

We can expect that, by applying similar methods to those of this thesis, more computationally sensitive proof systems could be discovered for temporal and counterfactual logics, and in some generality. Whether this is in fact so remains the major problem left open by this thesis. However, the avenue of attack is clear. In this chapter I prove correct a proof system that can be applied to logics such as temporal logics. This system is closely analogous to that of Chapter 4: indeed the system of Chapter 4 is a special case of the new system. The next stage, left undone here, is to apply arguments similar to those of Chapters 5 to 8 to the proof systems introduced in this chapter.

Section 9.1: More General Expansion Rules for Restricted Quantification

In Section 4.1 I introduced the following expansion rules and side conditions:

Quantifier Expansion Rules

(γ)	$\dfrac{\gamma}{\gamma_0(t)}$	(δ)	$\dfrac{\delta}{\begin{array}{c}\delta_0(a)\\ \delta_r(a)\end{array}}$

Side Conditions

(*1*) applies to δ-expansions:

a is a new name that has not appeared on the branch up to the point of the δ-expansion.

(*2*) applies to γ-expansions:

$\mathbf{R} \models_\Sigma \gamma_r(t)$ where $\mathbf{R}$ is the set of Σ-atoms on the branch up to the point of the γ-expansion.

(*3*) applies to γ-expansions:

t contains only names that have been introduced by δ-expansions on the branch up to the γ-expansion, except possibly the name ε.

(*4*)

There is a well defined and one to one function $f{:}\mathbf{P}^\delta \rightarrow \mathit{Names}$ such that $f(\delta)$ is the name introduced by into any immediate subposition of δ. That is, for each δ-type position $\delta \in \mathbf{P}$, if $\delta_0(a) \in \mathbf{P}$ then $f(\delta) = a$.

I showed in Chapter 4 that incorporating these rules into a sound and complete propositional expansion system **S** led to a sound and complete expansion system $\Sigma\mathbf{S}(1234)$ for restricted quantification, *if* Σ satisfies the least Herbrand model property. Also, in Section 4.5, I showed that we only need to use Σ-closure instead of closure for sound and complete systems if Σ satisfies the weak least Herbrand model property. Now I consider what happens if neither of these two conditions are satisfied. In fact it is only necessary to replace the rule (γ) and its associated side condition (*2*).

New Quantifier Expansion Rule (γ-branch)

$$(\gamma\text{-branch})\quad \frac{\begin{array}{c}\gamma^1\\ \vdots \\ \gamma^n\end{array}}{\begin{array}{c}\gamma_r^1(t_1)\\ \gamma_0^1(t_1)\end{array} \;\Big|\; \cdots \;\Big|\; \begin{array}{c}\gamma_r^n(t_n)\\ \gamma_0^n(t_n)\end{array}}$$

Notation

If **S** *is any propositional expansion system, I will use "*Σ**branchS***" to refer to the expansion system containing the expansion rules of* **S***, as well as (γ-branch) and (δ).*

The side condition which will replace (*2*) is defined below.

Definition 9.1.1 (Side Condition (*6*)**)**

(*6*) applies to γ-expansions:

$\mathbf{R} \models_\Sigma \gamma_r^1(t_1) \vee \cdots \vee \gamma_r^n(t_n)$ where $\mathbf{R}$ is the set of Σ-atoms on the branch up to the point of the γ-expansion.

There are two immediate points to note about the new rule.

First, it is extremely unpleasant compared to the simple rule (γ). A single application can involve arbitrary (finite) branching. Also, we must explicitly introduce the subformulas $\gamma_r^i(t_i)$, which before were never introduced onto the tree. Furthermore, checking the side condition (*6*) involves checking the provability of a disjunction, effectively meaning that arbitrary reasoning may be involved rather than just reasoning within least Herbrand models.

The second point is that the old rule is a special case of the new one, given the completeness result I will show later in this chapter. Consider any application of (γ-branch). If Σ satisfies the least Herbrand model property, and $\mathbf{R} \models_\Sigma \gamma_r^1(t_1) \vee \cdots \vee \gamma_r^n(t_n)$, then for some i, $\mathbf{R} \models_\Sigma \gamma_r^i(t_i)$. Therefore we need not consider any of the other branches, and because $\mathbf{R} \models_\Sigma \gamma_r^i(t_i)$, we may delete $\gamma_r^i(t_i)$ from the ith branch. So we can make any application of (γ-branch) into an application of (γ) without losing completeness, *if* Σ satisfies the least Herbrand model property.

Section 9.2: Example

Temporal logics are playing an increasing role in computer science and artificial intelligence. Unfortunately, the accessibility theories for temporal logics tend to fail the least Herbrand model property. Indeed, there are cases where the theory cannot be axiomatised at all in the predicate calculus, for example temporal logics in which the set of times is the set of integers. However, if the accessibility theory can be expressed by the predicate calculus then the proof systems introduced in this chapter can be used. An example of such a logic is the modal logic **S4.3.**

The semantics of the accessibility theory of **S4.3** are those of **S4**, that it must be reflexive and transitive, with the additional demand that the accessibility relation must be *weakly connected.* This condition is expressed by the predicate calculus formula

$$\forall w\ \forall x\ \forall y\ (w \mathcal{R} x \wedge w \mathcal{R} y) \supset (x \mathcal{R} y \vee y \mathcal{R} x)$$

and by the *non-Horn* clause

$$w \mathcal{R} x, w \mathcal{R} y \ \rightarrow\ x \mathcal{R} y, y \mathcal{R} x$$

Intuitively, this condition demands that given any two accessible worlds, one must "come second" in the sense of being itself accessible from the other one. The word "weakly" in "weakly connected" indicates that two distinct worlds are allowed to be mutually accessible. Given a weakly connected accessibility relation, the set of worlds can be seen as a linear sequence of clusters with the property that all the worlds in each cluster are mutually accessible.

The axiom added to **S4** to yield **S4.3** is

$$\Box(\Box A \supset B) \vee \Box(\Box B \supset A)$$

As an example of the use of the new (γ-branch) rule I will show how this formula can be verified given a weakly connected accessibility theory. The following is a **ΣbranchS**(*1346*) proof of the above formula.

$$\begin{array}{c}
\varepsilon \Vdash \mathbf{0}\Box(\Box A \supset B) \vee \Box(\Box B \supset A) \\
\hline
\varepsilon \Vdash \mathbf{0}\Box(\Box A \supset B) \\
\varepsilon \Vdash \mathbf{0}\Box(\Box B \supset A) \\
\hline
a \Vdash \mathbf{0}\Box A \supset B \\
\varepsilon \mathcal{R} a \\
\hline
a \Vdash \mathbf{1}\Box A \\
a \Vdash \mathbf{0}B \\
\hline
b \Vdash \mathbf{0}\Box B \supset A \\
\varepsilon \mathcal{R} b \\
\hline
b \Vdash \mathbf{1}\Box B \\
b \Vdash \mathbf{0}A \\
\hline
\begin{array}{cc}
a \mathcal{R} b & b \mathcal{R} a \\
\underline{b \Vdash \mathbf{1}A} & \underline{a \Vdash \mathbf{1}B} \\
\otimes & \otimes
\end{array}
\end{array} \quad (i)$$

(i) The expansion at (i) is an example of (γ-branch). It is applied to the two formulas $a \Vdash \mathbf{1}\Box A$ and $b \Vdash \mathbf{1}\Box B$. The side condition (*6*) is satisfied because $\varepsilon \mathcal{R} a$ and $\varepsilon \mathcal{R} b$ both appear on the branch and $\varepsilon \mathcal{R} a, \varepsilon \mathcal{R} b \vDash_{\Sigma} a \mathcal{R} b \vee b \mathcal{R} a$ (given that $\mathcal{R}$ must be weakly connected).

Goré (1991) has studied sequent style proof systems for temporal logics including **S4.3**, presenting systems which do not introduce any explicit syntax for times. There is certainly a close connection between Goré's system and the system I present here. In particular, as here, expansion of necessity type formulas can result in the splitting of a branch. As here, this splitting must be finite but can introduce arbitrarily many new branches. The same comments apply to de Swart's (1983) system for VCS and my system for VC (Gent 1990). It is plausible that these proof systems avoiding the use of explicit syntax for possible worlds and times could be derived from the system presented here, but it is difficult to formalise and generalise this intuition.

Section 9.3: Soundness of ΣbranchS(*16*)

I have already proved the soundness of the rule (δ) in Section 4.3. So, for the soundness of Σ**branchS**(*16*) given a sound propositional system **S**, I need only prove the soundness of the rule (γ-branch) when the side condition (*6*) is satisfied. The definition of a *sound* expansion rule is in Section 3.3 and the definition of the semantics of restricted quantification is in Section 2.3.

Theorem 9.3.1: Soundness of (γ-branch)

(γ-branch) is a sound expansion rule if the side condition (*6*) *holds.*

Proof

Suppose we have a set S containing formulas $\gamma^1, \ldots, \gamma^n$ that may be expanded under the rule (γ-branch) to yield $\gamma_0^1(t_1), \ldots, \gamma_0^n(t_n)$. Suppose also that S is satisfiable, and in particular that it is satisfiable in a Σ-model $\mathbf{M} = \langle \mathbf{D},\mathbf{I} \rangle$ under an assignment **A**. That is, for each i, $[\gamma^i]^{\mathbf{I},\mathbf{A}} = \mathbf{t}$.

By the side condition (*6*), $\mathbf{R} \vDash_\Sigma \gamma_r^1(t_1) \vee \cdots \vee \gamma_r^n(t_n)$, where **R** is the set of restrictions in S. Since **M** is a Σ-model of S, **M** is a Σ-model of **R** and hence of $\gamma_r^1(t_1) \vee \cdots \vee \gamma_r^n(t_n)$. But by the definition of a model, for some i, **M** is a Σ-model of $\gamma_r^i(t_i)$. That is, $[\gamma_r^i(t_i)]^{\mathbf{I},\mathbf{A}} = \mathbf{t}$. But the definition of semantics of restricted quantifiers, and the fact that $[\gamma^i]^{\mathbf{I},\mathbf{A}} = \mathbf{t}$, ensure that $[\gamma_0^i(t_i)]^{\mathbf{I},\mathbf{A}} = \mathbf{t}$. Therefore **M** and **A** satisfy S, $\gamma_r^i(t_i)$, $\gamma_0^i(t_i)$. That is, the set of formulas on one of the expanded branches is satisfiable, which was what was wanted.

□

Section 9.4: Completeness of ΣbranchS(*1346*)

Proving completeness of the new proof system is not difficult, but slightly more fiddly than the proofs in Sections 4.4 and 4.5 because now there will not be a single model for a branch $\mathbf{M}_\lambda$. However, at least all that is needed is a proof of the relevant formulation of the Branch Model Lemma. After that, exactly the same completeness proof as I used in Section 4.4 will do. I should refer you back to one definition used here which I have not used since Chapter 4. This is the definition of Σ-closure (Definition 4.5.1). For convenience I repeat the rather more complicated definition of a model for a branch.

Definition 4.4.2.1 ($\mathbf{M}_\lambda$, Σ-model for a branch)

Suppose that λ is a branch in a Σ**S** tree, containing the set of formulas U and set of restrictions **R**. In that case, $\mathbf{M}_\lambda = \langle \mathbf{D},\mathbf{I} \rangle$ is a Σ-*model for the branch* λ if:

1. **D** is the set of all terms built using function symbols from the constants and names that appear anywhere in U; and
2. **I** applied to constant symbols and names is the identity function; and
3. **I** applied to a n-place function symbol f gives the function defined by

$$f^{\mathbf{I}}(t_1, \ldots t_n) = f(t_1, \ldots t_n)$$

4. for any γ-type formula γ on the branch, and any term t arising on the branch, $\mathbf{M}_\lambda$ satisfies the associated restriction $\gamma_r(t)$ if and only if $\mathbf{R} \vDash_\Sigma \gamma_r(t)$; and
5. for any P, an $\mathcal{L}$-atom, **M** satisfies P if and only if $\mathbf{1}P \in U$; and
6. $\mathbf{M}_\lambda$ is a Σ-model.

In the following proof I will use the fact that if a set of restrictions is Σ-unsatisfiable then a finite subset of them is Σ-unsatisfiable. This follows from the compactness of first order logic (see for instance Smullyan 1968) and because Σ-unsatisfiability is a judgement of first order logic (recall that I only consider restriction theories Σ that can be expressed in first order logic).

Notice that for this theorem the notion of Σ-closure must be used (as defined in Section 4.5) rather than the usual notion of simple closure.

Lemma 9.4.1: Branch Model Lemma for ΣS(*1346*) with Σ-closure

If λ is an open (not Σ-closed) complete branch , containing the set of restrictions **R**, in an Σ**S**(*1346*) tree then there is a Σ-model $\mathbf{M}_\lambda$ for the branch λ.

Proof

We can certainly find a first order logic model $\mathbf{M}_\lambda$ satisfying parts (1) to (5) of Definition 4.4.2.1 since any first order logic model is well defined given the definition of which atoms hold in it. The only difficulty is to show that $\mathbf{M}_\lambda$ must be a Σ-model, to satisfy part (6) of the definition. I show that the supposition that $\mathbf{M}_\lambda$ is not a Σ-model leads to a contradiction.

Suppose $\mathbf{M}_\lambda$ is not a Σ-model. Then there must be a set of restrictions $\mathbf{R}'$ such that:

no element of $\mathbf{R}'$ appears on the branch; and

each Σ-model containing $\mathbf{M}_\lambda$ contains at least one element of $\mathbf{R}'$; and

each element of $\mathbf{R}'$ is an instance of $\gamma_r(t)$ for some position γ and term t arising on the branch.

This implies both that $\mathbf{R} \not\models_\Sigma \gamma_r(t)$ for each $\gamma_r(t) \in \mathbf{R}'$ and that $\mathbf{R} \cup \{\neg\gamma_r(t) \mid \gamma_r(t) \in \mathbf{R}'\}$ is Σ-unsatisfiable. Therefore, by compactness, for some finite subset $\{\gamma_r^1(t_1), \ldots, \gamma_r^n(t_n)\} \subseteq \mathbf{R}'$, $\mathbf{R} \models_\Sigma \gamma_r^1(t_1) \vee \cdots \vee \gamma_r^n(t_n)$. But then the condition (*6*) is satisfied by an application of the (γ-branch) expansion rule to $\gamma^1, \ldots, \gamma^n$. Since the branch is assumed to be complete, this application must have been applied somewhere on the branch, implying that for some i, $\gamma_r^i(t_i)$ appears on the branch and in **R**. But we assumed that $\gamma_r^i(t_i) \in \mathbf{R}'$ and that no element of $\mathbf{R}'$ appears on the branch. This is a contradiction, as promised.

□

Theorem 9.4.2: Completeness of ΣS(*1346*) with Σ-closure

If **S** *is a propositional countermodel expansion system, and a finite set of formulas S is not satisfiable, then there is a Σ-closed Σ**S**(1346)-tree for S.*

Proof

The proof is identical to the proof of Theorem 4.4.2.3.

□

As discussed at the end of Section 9.1, proof systems with the rule (γ) can be viewed as special cases of the same proof systems with (γ-branch). Indeed, when the least Herbrand model property holds, the simple argument given earlier converts easily into a completeness proof for systems with (γ). Indeed, in my technical report (Gent 1991b) I present the system of this chapter first, regarding the system of Chapter 4 as a corollary. On the other hand, most of this thesis is devoted on proof systems which use (γ) and therefore it is more natural to present them independently, as is done in Chapter 4.

Chapter 10

Related Work, Further Work, and Conclusions

In this chapter I put the work in this thesis in context, first by considering related work from the literature, and then by making suggestions for future research based on this thesis. Finally, I give some conclusions to the whole thesis.

Section 10.1: Related Work

In this Section discuss how other work in the literature relates to the work presented in this thesis. I structure this discussion by considering how other work relates to the aims, methods, and results of this thesis.

Section 10.1.1: Work with Related Aims

In this thesis my aim has been to study the link between proof systems for quantified logics and proof systems for the underlying propositional logic. I have done this in the context of restricted quantification, which easily expresses a variety of sorted and modal logics. I have tried to consider as wide a range as possible of propositional proof systems, only restricting the range of propositional systems when this was theoretically necessary.

Of course a lot of work has related quantified proof systems and propositional proof systems: almost any work on proof systems for first order logic can be seen in this light. However, very little work has both addressed proof systems for restricted quantification, and sought to do this without concentrating on a particular propositional proof system. The only significant earlier work doing this is by Frisch (1989, 1991) and Scherl (1991, Frisch and Scherl 1990, 1991).

Frisch (1989, 1991) presents ''a general framework for sorted deduction''. The framework is very similar to the framework I have used in this thesis: restricted quantification is used in the presence of a restriction theory. Furthermore, showing clearly how close our work is, Frisch requires the least Herbrand model property of the restriction theory exactly as I have done throughout the major part of this thesis, starting in Chapter 4. Frisch's restrictions must be monadic, as opposed to restrictions of arbitrary arity that I allow. However, there would seem to be no significant problem extending Frisch's approach to polyadic restrictions. Frisch claims that his framework ''provides for a *systematic* transformation of unsorted deductive systems into sorted ones'' (Frisch 1991, abstract). Thus Frisch does not commit himself to one particular propositional proof system, and in this rests the crucial similarity with my own aims. Frisch and Scherl (1990, 1991) show how to extend the approach to deal with modal logics.

An important feature of Frisch and Scherl's work, that has no parallel in my own thesis, is their demonstration of the application of constraint based reasoning to modal logics (Frisch and Scherl 1990, 1991). In particular, Frisch and Scherl show in a general way how the use of sequences to represent possible worlds arises, explaining a common feature of much work on theorem proving in modal logics, such as Fitting's (1983, Chapter 8) prefixed tableau, and the work of Wallen (1987b, 1989) and Ohlbach (1988a, 1988b). In Appendix 1 I show how sequences can be seen to arise in modal logics, based on the work in this thesis.

Where I have ignored the problems involved in unification, Frisch and Scherl put this centre stage. In particular, special purpose unification is required for each different restriction theory, since Frisch shows how to reduce reasoning within the theory to just this special purpose unification. This certainly makes their work more immediately useful if one wishes to implement a theorem prover, since unification is certainly essential to efficient first order theorem proving. On the other hand, for a clear theoretical

understanding of the issues involved in lifting propositional proof systems to restricted quantification, I see two theoretical advantages to my approach. First, I achieve a greater generality than Frisch. Frisch is restricted to analogues of quantified proof systems in which existential quantifiers are skolemised. In this thesis I gave matrix characterisations of validity for restricted quantification using an acyclicity test instead of skolemisation. Second, my use of ''expansion systems'' has enabled me to give formal proofs of the correctness of extended systems, and precise conditions for correctness to hold, in terms of properties of the expansion system and the restriction theory. This contrasts with Frisch's claim that a proof of correctness of a proof system for unrestricted quantification can be systematically transformed to become a proof of correctness of a proof system for restricted quantification: for instance his correctness proof for sorted resolution (1991, section 6) is presented as being merely an *example*. Frisch does not present a general, formal, proof of his claim. This lessens the ease of applying his result, since it might not be clear if a particular correctness proof could be transformed in this way.

Although I have not considered unification, at least my work shows clearly how special purpose unification will arise. First, one views unification as a syntactic operation, simply delaying a firm choice as to the introduction of specific terms (a view expounded in Fitting (1990b), for example). Unification arises out of universal instantiation. But that is exactly where I attached special purpose reasoning in the proof systems of Chapters 4, 5, 7, and 8. Special purpose unification arises as the combination of special purpose reasoning and the desire to delay the choice of term to introduce.

Tony Cohn and I have discussed some of these issues further in (Gent and Cohn 1991).

Section 10.1.2: Work using Related Methods

There are two aspects of the methods I have used in this thesis that I wish to discuss. The first is the methodology used to derive the proof theoretic results in this thesis. The second is the approach of theorem proving in modal logics by translating modal logics into first order logic, by expressing Kripke semantics.

The methodology used in this thesis is to derive computationally sensitive, and possibly complex, characterisations of validity by proof theoretic arguments applied to a much simpler, but possibly computationally unsuitable, proof system. These theoretical arguments aid understanding of the relationship between different proof systems. The arguments apply because the expansion rules introduced in Chapter 4 are particularly well suited to this development, the reason of course why the rules were introduced. This methodology avoids the need for many semantic arguments, which are often infinitary, replacing them with arguments applied to proofs, proofs being finite syntactic objects. This methodology I have used in this thesis was introduced to me by Wallen, both in person and by his work.

The methodology is well exemplified by Wallen (1986, 1989) who shows how to develop matrix based characterisations of validity from tableau systems. Interestingly, Wallen's formal proofs of matrix characterisations for modal logics (1989, Chapter 5) are direct from the semantics of modal logics. In a sense, then, my formal proofs in Chapter 7 are new, but the arguments I use in that chapter are entirely due to Wallen. Another example of the methodology is given by Pym and Wallen (1990).

My dependence on the earlier arguments of Wallen is particularly clear in the development of Chapters 5 and 7. In Chapter 5 I address the problem of reducing some of the order dependence between applications of expansion rules which arises from theory reasoning: eliminating order dependence was shown to be important by Wallen. In Chapter 7 I exploit the fact that all order dependence arising from the theory has been eliminated. I am able to apply Wallen's arguments to derive a matrix characterisation of validity.

The fact that I use Wallen's arguments does not mean that the material in Chapter 7 is not new. My decision to study restricted quantification separates me from Wallen and generalises his approach to a wider class of logics than he considered in (Wallen 1989). Expressing modal logics using restricted quantification makes it quite clear where the extra syntax for dealing with modalities comes from and

enables me to apply Wallen's insights more directly than he did himself.

Mints (1988, 1990) has also used explicit transformation between proof systems to derive new proof systems. He derives resolution style proof systems for modal and intuitionistic logic from sequent style proof systems. The results are not directly related to mine because of his use of implicit ways of dealing with modalities rather than the explicit introduction of names for worlds I have used. Nevertheless, his elegant results are another justification of the methodology I apply in this thesis, since a number of other resolution systems for modal logics using implicit methods for modalities are all inelegant or use unsatisfactory analogues of resolution. Such systems have been proposed by Abadi and Manna (1986), Chan (1987), Geissler and Konolige (1987), Fariñas del Cerro and Herzig (1988), Enjalbert and Fariñas del Cerro (1989), and Cialdea (1991). Only Fitting's (1990a) ''destructive modal resolution'' systems, not derived by analysis of other proof systems, can compare with Mints' systems for elegance and closeness to classical resolution.

In this thesis I have particularly emphasised the application of my work to modal logics. The approach I have taken to theorem proving in modal logics is to translate modal logics into logics of restricted quantification, the restriction theory being chosen according to the conditions on the modal accessibility relation. Most work concerning theorem proving in modal logics takes a different approach, using special purpose arguments for modalities and possible worlds. Examples include all the methods discussed in the previous paragraph, as well as all the methods of Fitting (1983) and Wallen (1989).

The first work considering explicit translation of Kripke semantics for automated theorem proving is that of Morgan (1976). Using a standard first order resolution theorem prover, Morgan was able to prove all the modal logic theorems in Hughes and Cresswell (1968). However, Morgan's approach has an obvious drawback, in that the full power of resolution theorem proving is brought to bear on reasoning in the accessibility theory. Many accessibility theories can be reasoned about very much more efficiently than by the undecidable method of resolution. This observation is central in this thesis, where all formulas from the accessibility theory go into restrictions on quantifications. Since restrictions are dealt with separately by all the proof methods in this thesis, we can take advantage of any simplicities in the accessibility theory.

Ohlbach (1988a, 1988b, 1989, 1990) has studied semantics based translation methods for modal logics. Compared to my work he considers a wider variety of translations than the simple translation using the accessibility relation I use in this thesis. Ohlbach (1990) studies in depth ''functional translations''. In these, a set of ''modal context access functions'' exist, the application of each function to a world returning some world accessible from the first. Since Ohlbach demands that functions must be total, the method is only well suited to serial modal logics. The conditions on the accessibility relation are replaced by conditions on the set of context access functions. For example, reflexivity is expressed by the demand that the identity function should be in the set of access functions. Translation of a modal formula uses ordinary quantification over context access functions. For example, the modal formula

$$\Diamond \Box (P \wedge \Diamond Q)$$

would be translated as

$$\exists f \, \forall i \, (P(\downarrow(f \circ i, 0)) \wedge \exists g \, Q(\downarrow(f \circ i \circ g, 0))).$$

In this expression "0" represents the initial world while "$\circ$" indicates function composition and the "$\downarrow$" terms indicate application of the function to the initial world.

The reason for the functional translation is that, as Ohlbach puts it (1990, page 4): ''Most of these properties can be described with equations, which, in a further step, can be translated into a theory unification algorithm.'' The same point is further discussed by Auffray and Enjalbert (1989).

One drawback of Ohlbach's work is that the target logics for his translation seem to me to be needlessly complex, making his results less clear and less easily extensible. For example, in (Ohlbach 1990) his target logic is Schmidt-Schauss' (1989) order sorted predicate logic, leading Ohlbach to say ''Since

Schmidt-Schauss' logic is quite complex, we briefly introduce its main notions.'' In this thesis I deliberately chose a very simple target logic, the logic of restricted quantification presented in Chapter 2. I was able very quickly to introduce *all* its notions. This makes the proof theory of the target logic much simpler even though it still provides an explanation for the use of special purpose reasoning in restriction theories. A linked drawback of Ohlbach's work is that he only considers resolution style proof systems for the target logic. Because of the complexity of the target logics producing analogues of other propositional proof methods would not necessarily be simple.

Fariñas del Cerro and Herzig (1990) translate modal logics into ''deterministic modal logics'', using a similar functional translation to Ohlbach's. It is not clear what the advantages are of their new formalism over and above the use of the functional translation.

Frisch and Scherl (1990, 1991), discussed also in Section 10.1.1 of this chapter, translate modal logics into a first order logic of restricted quantification. This enables them to use insights from constraint reasoning, illustrating how sequence unification arises in modal logic theorem proving. A sequence is very similar to Ohlbach's chain of functions. Their work has the advantage over Ohlbach's that they use a much simpler target logic, and are not restricted to resolution based proof systems. As I mentioned earlier, however, they are restricted to extending first order proof systems that rely on skolemisation.

The work of Ohlbach and Frisch and Scherl, as well as the work in this thesis, shows that translating modal logics into first order logics still allows careful use to be made of the difference between modalities and ordinary quantifiers, a distinction lost in methods such as Morgan's. That is, we provide special purpose reasoning methods for modal logics using general considerations. Other methods have been proposed using explicit syntax for possible worlds, but using special purpose arguments for modalities, for example by Wallen and Jackson and Reichgelt (discussed in Section 10.1.3). These proposals do not seem to contain any insights that cannot be duplicated in a more general framework such as mine.

Section 10.1.3: Work producing Related Results

In this section I survey work from the literature whose results are related to results I have produced in this thesis. I will do this by working through my thesis chapter by chapter, discussing work relating to each chapter. For convenience I provide subheadings indicating which chapter I am discussing.

Chapter 2

There is no new work in Chapter 2. I introduce the formalism I use for restricted quantification as well as an extension of Smullyan's (1968) uniform notation for it, and discuss how to express modal logics using restricted quantification.

The first work I know of discussing restricted quantification is by Hailperin (1957a, 1957b). Hailperin does not consider automated theorem proving, and does not separate syntactically what may appear as restrictions and what may not: indeed arbitrary formulas of the predicate calculus may appear. Because of this lack of separation Hailperin does not include a separate restriction theory. This idea may be found, for example, in Frisch (1991) or Bürckert (1990a).

Two aspects of my formalism are worth brief discussion. Unlike Hailperin I demand that all restrictions must be atoms rather than arbitrary formulas. It would be perhaps more elegant to allow arbitrary formulas[1] and it would make little difference to the work of Chapters 4 and 9. However, knowing that restrictions are atoms is convenient for the work of Chapters 5 to 8. The other limitation of my formalism is that formulas from the restriction theory may only appear as restrictions. This is to force the separation between the restriction theory and the rest of the logic that is the subject of this thesis. However, Hailperin makes no such demand, and Cohn (1987, 1989b) has explicitly argued, in the context of many sorted logic, that one should allow restrictions to appear anywhere.

[1] and I thank Felix Hovsepian for pointing this out to me.

Other formalisms for logics than restricted quantification could be used to study the relation between propositional and quantified proof systems. An example is Smullyan's (1970) ''Abstract Quantification Theory''. However, the particular advantage of using restricted quantification is that the notation gives explicit access to the underlying theory, which would not be true, for example, if we applied abstract quantification theory to modal logics.

Chapter 3

In Chapter 3 only two very minor new pieces of work are discussed. The first is my definition of ''countermodel expansion system'', a definition which is new. As noted there, this refines slightly the notion of an ''expansion system'' due to D'Agostino (1990). The usefulness of the definition is shown in Chapter 4. The second new work in Chapter 3 is my extension of **KE** to deal with equivalences. This is a case of the obvious thing working. The new rules are very **KE**-like and it is a very natural extension, unlike the usual extension of analytic tableau to deal with equivalences.

Chapter 4

In Chapter 4 I showed how to extend any propositional countermodel expansion system to deal with restricted quantification, provided that the restriction theory has the least Herbrand model property. I know of no work which extends such a variety of propositional proof systems in the presence of such a general condition. For example, Frisch (1989, 1991) requires further properties of restrictions, akin to my non-vacuity condition. Also, Frisch extends existing first order proof systems rather than propositional ones.

The condition the least Herbrand model property is required is not new. Similar results for proof systems of restricted quantification have been presented by Frisch (1989, 1991) and Bürckert (1990a). Recall that the least Herbrand model property is guaranteed if the restriction theory can be written in Horn clauses without goal clauses (Theorem 4.4.1.4). That Horn clauses are important to the system of Chapter 4, and hence to all the work presented throughout Chapters 5, 6, 7, and 8, represents further enforcement of the fact that Horn clauses are essential to computation. As well as the close relationship between Horn clause logic and intuitionistic logic, Horn clauses are fundamental to logic programming and Prolog (see for example Lloyd 1984). Furthermore, they play a crucial role in algebraic specification in theoretical computer science (see for example Wirsing 1990).

A feature of the work in Chapter 4 is the use of an arbitrary propositional countermodel expansion system as the base. Some work has been done which is related if one fixes propositional analytic tableau as the base proof system. There is a very close relationship with Fitting's (1972, 1983 Chapter 8) ''prefixed tableaux'' for modal logics. Prefixes play the role of possible worlds, and thus Fitting has introduced explicit syntax for possible worlds. Fitting gives prefixed tableaux systems for a wider range of modal logics than I consider, or could easily consider, since he presents systems for non-normal modal logics.[2] For those modal logics I can deal with, Fitting's systems could be very easily derived from my system, apart from one feature. This is Fitting's use of sequences as prefixes. A simple derivation of prefixed tableaux from my system of Chapter 4 would use single names for worlds. The difference can be illustrated as follows. In Chapter 4, from a signed formula $\mathbf{1}\, w \Vdash \Diamond P$ one would introduce $\mathbf{1}\, w' \Vdash P$ where w is some name and w' is a new name. Fitting however, introduces from $\mathbf{1}\, p \Vdash \Diamond P$ the formula $\mathbf{1}\, pw' \Vdash P$ where p is some sequence of names and pw' is some new sequence of names (whether or not w' is new being unimportant). Then, instead of explicitly referring to first order deductions about the accessibility theory, Fitting defines the accessibility relation on the set of prefixes by saying that $q \mathcal{R} qw$ for any sequence q and name w, and adds the tableau rule

$$\frac{\mathbf{1}\, p \Vdash \Box P}{\mathbf{1}\, q \Vdash P} \quad \textit{where } q \textit{ is a sequence such that } p \mathcal{R} q.$$

[2] Non-normal modal logics are those in which the set of formulas true in a world is not forced to be consistent.

Of course the accessibility relation is defined to satisfy extra conditions appropriate to the logic. For a longer introduction to prefixed tableaux, and a sketch proof of their soundness and completeness based on the work up to Chapter 6 in this thesis, see Appendix 1. Catach (1991) describes an implementation of prefixed tableaux-like systems for a variety of propositional modal logics, using a method due to Hughes and Cresswell (1968) to make the systems decidable.

Introducing sequences as prefixes provides a simple representation for the least Herbrand model that one would naturally use in my method. Otherwise, there seems to be little to choose between the two methods for humans constructing proofs. However, the use of sequences to represent worlds in modal logics seems to be important for automated theorem proving, as shown by Frisch and Scherl (1990, 1991).

Schmitt and Wernecke (1990) present a tableau proof system for order sorted logic. They present a ground proof system (that is, only introducing explicit terms containing no variables) and then discuss how to lift it to a system with unification. Their ground system is very easily seen to be an instance of the proof systems I present in Chapter 4, and the lifting to unification is standard. The same remarks apply to the tableau system presented by Costa and Cunningham (1991) for their action logic. It is interesting to observe that Schmitt and Wernecke, by treating order sorted logic as a special case, have missed the wood for the trees: they propose as a research problem ‘‘What is an adequate general framework for establishing results (e.g. soundness and completeness) for variants of many sorted predicate logic?.’’ This is a problem that Chapter 4 solves.

Chapters 5 and 6

In Chapter 5 I showed how it was possible to reduce the amount of order dependence of rule applications in proof trees, by introducing new rules for restricted quantifiers. I defined two new conditions on the restriction theory, alphabetical monotonicity and non-vacuity, which I used to show the soundness of the resulting proof systems. In Chapter 6 I showed how to prove these new conditions for a variety of restriction theories. The new work in Chapters 5 and 6 is not as important as the use I put it to later.

The importance of alphabetical monotonicity and non-vacuity is their use in eliminating order independence of rule applications. It is this that allows the derived proof systems of Chapters 7 and 8 to be constructed, in which propositional and quantificational reasoning are largely separated. My presentation isolates clearly the process of reducing order independence, which in itself is a contribution. The only change to the expansion rules in Chapter 5 is in the side condition on the (γ) rule, and this change clearly reduces order dependence in the proof system. In Chapter 6, where I discuss proving alphabetical monotonicity and non-vacuity, I do not discuss proof systems at all. That is, the conditions which enable us to derive new proof systems can be checked without any reference to those proof systems.

I know of nothing similar to the alphabetical monotonicity condition in the literature. On the other hand, the non-vacuity condition is very similar to Frisch’s (1989, 1991) condition called "Σ-satisfiability". Frisch’s condition also plays a very similar role to non-vacuity. Σ-satisfiability is required of restrictions[3] for skolemisation to be valid, and skolemisation is the process that separates quantificational and propositional reasoning. I make more use of the non-vacuity condition than Frisch does of Σ-satisfiability, using it to derive matrix characterisations of validity in Chapter 7 using acyclicity tests rather than skolemisation. Hailperin(1957b) also mentioned the necessity of disallowing free variables with vacuous ranges in the context of restricted quantification.

[3] Non-vacuity is a condition on Σ given a set of restrictions, while Σ-satisfiability is a condition on restrictions given Σ. This reversal is unimportant.

Chapter 7

In Chapter 7 I derive matrix characterisations of validity in the presence either of non-vacuity or of alphabetical monotonicity.

As I discussed in Section 10.1.2 in this chapter, the arguments I use in Chapter 7 are due to Wallen (1986, 1989). This does not mean that my results are not new. Wallen's arguments were proposed as means of deriving matrix characterisations of validity from tableau-style calculi. This is precisely the use I put these arguments to. The crucial decision is which tableau-style system to apply the arguments to.[4] The extension of Wallen's arguments to arbitrary expansion systems in Chapter 7 is not technically difficult since expansion systems are tableau-style systems.

Wallen (1987b, 1989) presents matrix characterisations of validity for various modal logics. Chapter 7 can be seen as a generalisation of his results in two independent directions. First, I derive analogues of matrices for expansion systems other than analytic tableau. Second, my matrix characterisations of validity apply to a much wider range of logics than Wallen's. Furthermore, it is only necessary to prove alphabetical monotonicity or non-vacuity to derive matrix characterisations for new logics. I do this in Chapter 6 for a variety of modal and sorted logics.

The work of Chapter 6 enables me to give matrix characterisations of validity for all the modal logics Wallen considers. The only logic he considers that I do not is the intuitionistic logic **J**. Since this can be expressed by a translation into **S4** or Kripke style semantics can be given for it directly, it would be easy for to cope with it given my results.

Unfortunately, the results of Chapter 7 do not allow Wallen's characterisations to be read off as simple corollaries.[5] The principal difference between our two characterisations for modal logics is Wallen's use of sequences as prefixes, rather than my use of names. Note that this distinction is the same as that I noted in comparison with Fitting's prefixed tableaux (see the discussion of Chapter 4 in this section). This is no coincidence. Indeed, Wallen states that historically his methods were inspired by Fitting's (Wallen 1989, page 189). This lack of coincidence is underlined by Frisch and Scherl's (1990, 1991) work showing how sequences arise in this context. The work on prefixed systems in Appendix 1 could be used as a basis for a new derivation of matrix characterisations of validity very similar to the derivation in Chapter 7: the result would be very much closer to Wallen's characterisations than that actually presented in Chapter 7.

Wallen (1989, Chapter 6) explicitly discusses unification problems, which I have not. This and his use of sequences certainly makes his work more immediately applicable. Indeed some of Wallen's characterisations have been used to implement an automated theorem prover for the modal logic **S5** (Wallen and Wilson 1987). Frisch and Scherl's work should make it easy to introduce sequences and sequence unification into the matrix characterisations I present in Chapter 7.

In a much longer discussion of theory resolution, Stickel (1985) briefly discusses a matrix characterisation of validity incorporating special purpose reasoning in a theory. If one used his approach for restricted quantification, then restrictions would explicitly appear in the matrix, thus greatly expanding the number of paths through a matrix and so the computational cost of checking them all for closure. This contrasts with matrices as presented in Chapter 7, where restrictions do not add any paths to a matrix, while my characterisation of validity still incorporates special purpose reasoning in the restriction theory.

Petermann (1989, 1990a, 1990b, 1991) has discussed incorporating theories into the connection method at much greater length than Stickel. In particular, he considers the problems of lifting Stickel's results to

[4] Indeed, providing the correct tableau-style system was the historical reason for the whole of the development of Chapters 4 to 6.

[5] From my study of this issue it seems that they could be read off as complicated corollaries, with extremely tedious and unilluminating proofs. They wouldn't really be corollaries then, would they?

allow free variables (Stickel's approach was limited to ground proof systems). Petermann also considers the problems of incorporating theories with equality, showing in (Petermann 1991) how to incorporate the methods of paramodulation (Wos *et al* 1967), E-resolution (Morris 1969) and RUE-resolution (Digricoli and Harrison 1986) into the connection method. Therefore Petermann's work represents a considerable advance on Stickel's brief consideration of this subject.

In comparison with my own work, Petermann's has the advantages of being able to deal with equality and of considering the problems of unification: like Stickel I deal only with ground based proof systems. Petermann's presentation is much closer to an implementation than mine because he centres his discussion on the connection method. Petermann's work should be applicable to other theorem proving methods since it is based on a generalisation of Herbrand's theorem. Unlike me, however, Petermann has not considered theorem proving for logics of restricted quantification, and this represents the main advantage of my work. That is, if formulas from the theory appear in a theorem to be proved, then those formulas will appear explicitly in a matrix. Hence, formulas restricting universal instantiation will expand the number of paths to be considered. This does not happen in my matrix characterisations of validity because restrictions need not appear explicitly in a matrix. However, it is important to observe that this kind of problem is not insuperable in every case. For example, Petermann (1990a, 1990b) shows how his work can be applied to modal logics. He does this using Auffray and Enjalbert's (1989) translation of modal logics into ''path-logic''. This translation is similar to Ohlbach's functional translation discussed in Section 10.1.2. Indeed path-logic is just a variant of first order logic. The advantage of using this translation is that modal formulas are translated into ordinary quantifiers rather than restricted quantifiers, and so modal connectives do not lead to any new paths. Different modal logics are obtained by using different theories about the functions. Interestingly, Petermann (personal communication 1991) points out that it should be possible to extend this approach to non-serial modal logics since his extension of Herbrand's theorem (1990b) takes account of partial functions.

In Chapter 7 I showed how Wallen's arguments could be applied starting from expansion systems other than tableau. It is not clear how useful this generalisation will be, since I have not investigated its usefulness in this context. The results could certainly be directly applied to the expansion system **KE**. In his thesis containing much study of **KE**, D'Agostino (1990, page 122) states that ''some variant of Bibel's connection method should prove useful. Similar proof-search procedures for non-classical logics could be developed by adapting Wallen's methods.'' The results in Chapter 7 confirm the claim of the second sentence, while leaving open the question of whether they would give a useful variant of the connection method.

Chapter 8

In Chapter 8 I showed how, if non-vacuity holds of a restriction theory, we can completely separate out quantifier instantiation from propositional reasoning. I did this by showing how applications of quantifier rules could permute freely with propositional rules.

Given the result of Chapter 8, any propositional proof system can easily be extended to a first order system. Furthermore, we can apply standard techniques for dealing with instantiation, such as skolemisation, or Bibel's (1987) reduction orderings.

Frisch (1991, Appendix B) presents what is effectively the same result as that of Chapter 8, for restriction theories in which each predicate is monadic. That is, he shows how skolemisation is possible given his condition called Σ-satisfiability. Demanding Σ-satisfiability is essentially the same as my own non-vacuity. Although showing the correctness of a skolemisation process is different to showing how to permute instantiation with propositional reasoning, the latter can be derived easily from the former simply by regarding ground instances of skolem terms as names. I believe that my presentation of the result puts it in a more fundamental light: the central importance is not that skolemisation works but that instantiation can be considered separately from propositional reasoning. Skolemisation is an important corollary of this fundamental fact.

Jackson and Reichgelt (1987, 1988, 1989) present resolution type methods for modal logics. If the sketch of skolemisation in Chapter 8 were to be given in full, then most of their systems could be read off as corollaries, except for the fact that Jackson and Reichgelt give explicit unification algorithms. However, this does not apply to the non-serial modal logics they consider, since the restriction theories for these logics do not satisfy non-vacuity, as required in Chapter 8. Unfortunately, Jackson and Reichgelt's contribution in this regard is difficult to understand because the difference between their systems for serial and non-serial modal logics lies only in the unification algorithm.

Chapter 9

Chapter 8 represents the end of the development I started in Chapter 4, the fundamental feature of which was that the restriction theory had to satisfy the least Herbrand Model Property. In Chapter 9 I consider a similar proof system to that of Chapter 4 for the case that Σ does not satisfy this property.

The system of Chapter 9 is unpleasant because universal instantiation introduces new branches in a proof tree, and because restrictions from universal statements appear explicitly on the new branches. This is analogous to the distinction between total theory resolution and partial theory resolution (Stickel 1985). In total theory resolution no new literals are introduced by resolution steps, while in partial theory resolution new literals, known as "residues", may appear.

Proof systems such as that of Chapter 9 have been little discussed in the literature. As well as Stickel's theory resolution, there is a close similarity with Bürckert's (1990a) resolution system for restricted quantification. Bürckert adds restrictions explicitly to clauses, conjoining the set of restrictions in two clauses when a binary resolution step is performed. To derive the empty clause is not enough, however. Instead one must derive enough empty clauses, each with an associated set of restrictions, such that each model of the restriction theory satisfies at least one of the sets of restrictions. This demand is very close to the rule (γ-branch) I present in Chapter 9. Instead of attaching a set of restrictions to each clause, one could achieve a similar effect by using different branches and labelling each branch with a set of restrictions and demanding that each model of the restriction theory satisfies the restriction on at least one branch. This is exactly the system of Chapter 9, except that I do not assume that we are dealing with formulas in clause form.[6] Bürckert's system is not a corollary of my general system since I need to assume we are dealing with a propositional countermodel expansion system: resolution is not a countermodel system. Just as I note that the expansion systems of Chapter 4 are special cases of that of Chapter 9 if the least Herbrand model property holds, Bürckert notes that only one empty clause is required in this case. Because of this, Bürckert's resolution system can be seen as a generalisation of Frisch's (1989, 1991) resolution system (although not of all Frisch's work since Bürckert only considers resolution).

There is a clear connection between the branching expansion system of Chapter 9 and some proof systems that have been proposed for modal logics whose accessibility theory cannot be written in Horn clauses. Examples include sequent-style systems for counterfactual logics such as those proposed by de Swart (1983) and me (Gent 1991), as well as similar systems proposed for modal logics such as **S4.3** by Goré (1991). All these systems have the common feature that some of the rules applied for modalities can cause branching, unlike the usual systems for the usual modal logics (see for example Fitting 1983). Unfortunately it is difficult to formalise this connection since the systems I have just mentioned deal with modalities using implicit methods without introducing names for worlds.

[6] Bürckert (1990a) does not discuss reduction to clause form or skolemisation, leaving this to his thesis (1990b).

Section 10.2: Further Work

In this section I propose some problems suitable for future research arising out of the work presented in this thesis. They can be summarised as follows:

- To extend the proof theoretical methods applied in the bulk of this thesis to restriction theories which fail the least Herbrand model property.
- To construct a good, general, implementation of the proof systems presented in this thesis, for the purpose of enabling researchers to build automated reasoners for a variety of logics very quickly.
- To investigate the value of matrix characterisations of validity based on expansion systems other than analytic tableaux.
- To apply similar techniques to those applied in this thesis to multiple-valued logics.
- To incorporate sequence unification into the proof systems presented in this thesis.
- To extend the proof systems presented in this thesis to deal with equality.

It is foolish to try to order either the importance or difficulty of problems that have yet to be solved. Instead I have ordered the points above by how much my work contributes to the *statement* of new research topics. In particular, expressing the first four points would have been difficult or impossible without the work in this thesis, as would appreciating that solutions to these problems can reasonably be hoped for by applying techniques applied in this thesis.

The most obvious issue left open in this thesis but requiring further work is suggested by the structure of this thesis itself. That is, in Chapter 4 I present a first extension of propositional expansion systems, and in Chapters 5 to 8 show how other proof systems can be derived from that system. However, the system of Chapter 4 is complete only if a certain condition holds, and in Chapter 9 I show how propositional expansion systems can be extended in a different way if the condition fails. Since the system of Chapter 4 is a special case of the system of Chapter 9, it is of great interest to know if the arguments of Chapters 5 to 8 can be seen as special cases of more general arguments. If so, then we could reasonably hope to use these arguments to produce good computational systems for logics not covered by the rest of this thesis. This problem is particularly important given that temporal logics are logics for which the system of Chapter 4 is incomplete.

A good implementation of the proof systems presented in this thesis, that took advantage of the generality of those proof systems, could significantly enhance future research. I certainly do not suggest that immediate efforts should be made to build world class theorem provers based on my work. Serious implementations of theorem provers tend to be the result of large teams and many years: for example the Markgraf Karl refutation procedure recently celebrated its fifteenth birthday (Ohlbach and Siekmann 1989). Also, I stand by the moral of Wallen's maxim quoted in Chapter 1, that implementations do not give a firm basis for choosing between different proof systems. However, there are two reasons why I believe implementations can be helpful. First, implementations can be easily distributed to allow other research workers to literally play with your ideas. In extreme cases they can be published: surely it is the length of the code (18 lines) as much as the good results that makes SATCHMO (Manthey and Bry 1988) so interesting? Second, results based on closely analogous implementations of related proof systems can be highly suggestive, especially where they tend to confirm theoretical arguments: Goré and D'Agostino (1991) provide just such a confirmation of D'Agostino's (1990) arguments on the relation between analytic tableaux and **KE**. I think that a well-written implementation of some of the proof systems in this thesis, allowing varying restriction theories and expansion systems to be incorporated, could allow faster research into the relationship between different proof systems for restricted quantification.

In Chapter 7 I defined an analogue of matrix characterisations of validity based on any expansion system. To date, the only such characterisation which has been shown to be computationally important is that based on analytic tableaux, which gives rise to the connection method (Wallen 1989, Bibel 1987).

Investigations of other matrix characterisations of validity have already been demanded by D'Agostino (1990), and for expansion systems such as **KE** could be of great importance, given D'Agostino's demonstration of the advantages of **KE** over analytic tableaux. However, it is not clear that important systems result based on other expansion systems, and therefore this topic requires further investigation. A good general implementation, as requested in the previous paragraph, would make investigating this much easier.

In this thesis I have given general arguments which are expressed in terms of restriction theories and expansion systems. However, I have used a fixed definition of polarities; namely that polarities are two-valued. Hähnle (1990, 1991) has shown that by allowing many-valued polarities one can derive tableaux proof systems for a large range of many-valued logics. As far as the proof theoretic arguments in this thesis go, there seems to be nothing crucial in my use of two-valued polarities. It should be quite straightforward to apply similar arguments in Hähnle's more general framework, thus deriving all the benefits obtained in this thesis for a much greater variety of logics. Many-valued logics are of great interest in computer science and artificial intelligence, as argued for instance by Belnap (1976, 1977). More recently Hovsepian (1992) has shown formally that some features of the world, such as the indistinguishability of close colours given imperfect sensors, can *only* be modelled accurately using many-valued logics.

Frisch and Scherl (1990, 1991) have shown in general how sequences (representing sequences of worlds) can be used in proof systems for modal logics. Once this is done, unification can be applied to sequences: in some cases special purpose unification algorithms eliminate all other reasoning in the restriction theory. Applying these techniques to my work should be straightforward, since Frisch and Scherl's presentation is quite general. As I discussed in Chapter 1, I have not considered unification in this thesis because it is not central to the proof theoretical arguments I apply: however it is certainly central to any efficient automated reasoning system.

In this thesis I have not considered logics containing equality: this is a significant drawback since very many applications of theorem proving require the use of equality, and equality provides some of the most difficult problems to the automation of reasoning. However, much of the theoretical work in this thesis, and certainly the approach I have taken, should be applicable to logics with equality: it should be natural to extend an approach involving theories to theories containing equality. This conjecture is supported by the fact that the connection method (Bibel 1987) can deal with equality, as can Petermann's (1991) work on incorporating theories into the connection method.

Finally, I wish to point out a minor technical problem that was left open in Chapter 5. That is, whether or not name-acyclicity is necessary for the soundness of $\Sigma\mathbf{S}(145)$ if Σ satisfies non-vacuity. A solution to this problem would increase my peace of mind, although I do not see that it is of any great importance.

Section 10.3: Conclusions

The first contribution of this thesis rests in the definition of a very simple way of building in the semantics of a restriction theory into a proof system, and the identification of general conditions under which the resulting proof system is complete. The resulting proof system has the triple advantages of being proved correct in a general way; of being a very natural extension of propositional proof systems, and so being easy for humans to use and reason with; and of laying a firm foundation for theoretical results about more complex proof systems. This last point is amply demonstrated by the fact that four chapters of this thesis are devoted to such analyses. I also define a second way of building in the semantics of a restriction theory, which is valid under more general conditions. This has the same three advantages. However, the last advantage is not demonstrated in this thesis since I do not undertake further analysis of the resulting proof systems.

The second contribution of this thesis is the identification of conditions under which order independence of rule applications can be reduced in proof systems for restricted quantification. In particular, I have

identified two conditions on restriction theories which enable a significant cause of order dependency in my first proof systems to be removed. I call these conditions ''alphabetical monotonicity'' and ''non-vacuity''. Non-vacuity is not an essentially new condition. In this thesis I show its importance proof-theoretically, showing that in its presence application of propositional proof rules can be completely separated from instantiation of restricted quantifiers. I also use non-vacuity to generalise existing techniques. My definition of alphabetical monotonicity is new, but in this thesis I show that even if alphabetical monotonicity holds, some order dependence remains in the resulting proof systems. This makes it much more desirable that non-vacuity should hold of a restriction theory than just alphabetical monotonicity.

Building on the first two contributions, I prove correct matrix characterisations of validity for restriction theories satisfying either alphabetical monotonicity or non-vacuity. The central motivation underlying matrix characterisations is the total elimination of order dependence of rule application and its replacement by global tests on resulting proofs. The test for the case where alphabetical monotonicity holds is more complex than that for non-vacuity, illustrating the greater usefulness of non-vacuity. These results represent the most significant single result in this thesis, since they generalise Wallen's matrix characterisations of validity for modal logics, both by extending the range of logics covered and by defining matrices more generally.

All the contributions in this thesis isolate and generalise insights which have earlier only been applied to particular systems or logics. I believe that this process of isolation and generalisation is vital to the development of automated reasoning as a science. As this process develops, more and more reasoning problems will become feasible using a combination of standard methods. Furthermore, new insights will be more immediately applicable. This is particularly important since building a good automated reasoning system is becoming a very complicated business: to facilitate building future systems it is essential that as much as possible of the experience of older systems should be able to be expressed both theoretically and in a way independent of the particular system.

Because I have sought to isolate a particular problem in this thesis, I have not addressed several issues crucial to the efficient implementation of automated theorem provers. The central problem I have considered is that of extending propositional proof systems to variable-free proof systems for restricted quantification, and the conditions on both the propositional system and the restriction theory under which the extension is correct. An example of a concept essential to efficient proof search that I have not considered is unification. This is not at all to minimise the importance of unification in automated reasoning: instead I simply claim that unification is not important to the problem I have addressed. Explicitly considering unification simultaneously with their proof system represents a significant drawback to the work of Jackson and Reichgelt (1987, 1988, 1989), to the extent that completeness proofs for their systems are not forthcoming (Jackson and Reichgelt 1989, page 190).

Suppose one wished to implement a theorem prover for a logic which could be expressed by restricted quantification. One might refer to work such as D'Agostino's (1990) concerning a suitable basic propositional proof system. By using the work in this thesis one could determine if and how this system could be extended to the full logic, depending on the properties of the restriction theory. By reference to work such as Frisch and Scherl (1990, 1991) one could determine how ideas derived from constraint reasoning might be applied, such as the use of sequences to represent possible worlds in modal logics. Work such as that of Siekmann (1986) on special purpose unification might enable reasoning within a restriction theory to be replaced or partially replaced by a special purpose unification algorithm. One can imagine, as I proposed in Section 10.2, a general purpose implementation allowing all of this work to be incorporated very easily. Even without this, it should be possible to produce very acceptable special purpose theorem provers while not needing to redo much of the theoretical work underlying the implementation.

References

To write a reference, you must have the work you're referring to in front of you. Do not rely on your memory. Do not rely on your memory. Just in case the idea ever occurred to you, do not rely on your memory.

-- Mary-Claire van Leunen, ''A Handbook for Scholars''

Abadi, Martin and Manna, Zohar, ''Modal theorem proving,'' in *8th International Conference on Automated Deduction*, ed. Jörg H Siekmann, pp. 170-189, Springer-Verlag, Berlin, 1986.

Andrews, Peter B, ''Theorem proving via general matings,'' *Journal of the Association for Computing Machinery*, vol. 28, pp. 193-214, 1981.

Atkinson, Will and Cunningham, Jim, ''Proving properties of a safety-critical system,'' *Software Engineering Journal*, vol. 6, no. 2, pp. 41-50, March 1991.

Auffray, Yves and Enjalbert, Patrice, ''Modal theorem proving: an equational viewpoint,'' in *Proceedings, International Joint Conference on Artificial Intelligence*, pp. 441-445, Morgan Kaufmann, 1989.

Belnap, N D, ''How a computer should think,'' in *Contemporary Aspects of Philosophy*, ed. G Ryle, pp. 30-56, Oriel Press, 1976.

Belnap, Nuel D, ''A useful four valued logic,'' in *Modern Uses of Multiple Valued Logics*, ed. J M Dunn and G Epstein, pp. 8-37, Reidel, 1977.

Beth, E W, ''On machines which prove theorems,'' *Simon Stevin Wissen Natur-kundig Tijdschrift* , vol. 32, pp. 49-60, 1958. (cited in D'Agostino (1990))

Bibel, Wolfgang, ''A comparative study of several proof procedures,'' *Artificial Intelligence*, vol. 18, pp. 269-293, 1982.

Bibel, Wolfgang, *Automated Theorem Proving,* Vieweg, second edition, 1987.

Boyer, R S and Moore, J S, ''The sharing of structure in theorem-proving programs,'' in *Machine Intelligence 7*, ed. B Meltzer and D Michie, pp. 101-116, Edinburgh University Press, 1972.

Bundy, Alan, *The Computer Modelling of Mathematical Reasoning,* Academic Press, London, 1983.

Bürckert, Hans-Jürgen, ''A resolution principle for clauses with constraints,'' in *10th International Conference on Automated Deduction*, ed. M E Stickel, pp. 178-192, Springer-Verlag, Berlin, 1990a.

Bürckert, Hans-Jürgen, *A Resolution Principle for a Logic with Restricted Quantifiers,* Dissertation, in preparation, 1990b. (cited in Bürckert 1990a)

Catach, Laurent, ''TABLEAUX: A general theorem prover for modal logics,'' *Journal of Automated Reasoning*, vol. 7, pp. 489-510, 1991.

Chan, Man-chung, ''The recursive resolution method for modal logic,'' *New Generation Computing*, vol. 5, pp. 155-183, 1987.

Chellas, Brian F, *Modal Logic: an Introduction,* Cambridge University Press, 1980.

Cialdea, Marta, ''Resolution for some first order modal systems,'' *Theoretical Computer Science*, vol. 85, pp. 213-229, 1991.

Cohn, A G, ''Taxonomic reasoning with many sorted logics,'' *Artificial Intelligence Review*, vol. 3, pp. 89-128, 1989a.

Cohn, A G, ''On the appearance of sortal literals: A non substitutional framework for hybrid reasoning,'' in *Proceedings, KR89: Principles of Knowledge Representation and Reasoning*, ed. R J Brachman, H Levesque and R Reiter, Morgan Kaufmann, Los Altos, 1989b.

Cohn, Anthony G, ''A more expressive formulation of many sorted logic,'' *Journal of Automated Reasoning*, vol. 3, pp. 113-200, 1987.

Cook, Stephen A and Reckhow, Robert A, ''The relative efficiency of propositional proof systems,'' *Journal of Symbolic Logic*, vol. 44, no. 1, pp. 36-50, 1979.

Cook, Stephen and Reckhow, Robert, ''On the lengths of proofs in the propositional calculus,'' in *Sixth Annual ACM Symposium on Theory of Computing*, pp. 135-148, Association for Computing Machinery, New York, 1974. (Corrections appear in *SIGACT News*, vol. 6 (1974), pages 15-22)

Costa, Marcos and Cunningham, Jim, ''Tableaux for an Action Logic,'' Medlar Esprit Basic Research Action Paper, 1991.

D'Agostino, Marcello, ''Investigations into the Complexity of Some Propositional Calculi,'' Technical Monograph PRG-88, Oxford University Computing Laboratory, 1990.

Digricoli, Vincent J and Harrison, Malcolm C, ''Equality based binary resolution,'' *Journal of the Association for Computing Machinery*, vol. 33, no. 2, pp. 253-289, April 1986.

Eisinger, Norbert and Ohlbach, Hans Jürgen, ''The foundations,'' in *Deduction Systems in Artificial Intelligence*, ed. Karl Hans Bläsius and Hans-Jürgen Bürckert , pp. 37-115, Ellis Horwood, Chichester, 1989.

Enderton, Herbert B, *A Mathematical Introduction to Logic,* Academic Press, New York, 1972.

Enjalbert, Patrice and Fariñas del Cerro, Luis, ''Modal Resolution in Clausal Form,'' *Theoretical Computer Science*, vol. 65, pp. 1-33, 1989.

Fariñas del Cerro, Luis and Herzig, Andreas, ''Linear modal deductions,'' in *9th International Conference on Automated Deduction*, ed. Ewing Lusk and Ross Overbeek, pp. 487-499, Springer-Verlag, Berlin, 1988.

Fariñas del Cerro, Luis and Herzig, Andreas, ''Deterministic modal logics for automated deduction,'' in *Proceedings, ECAI 1990*, pp. 262-267, 1990.

Fitting, Melvin, ''Tableau methods of proof for modal logics ,'' *Notre Dame Journal of Formal Logic*, vol. 13, pp. 237-247, 1972.

Fitting, Melvin, *Proof Methods for Modal and Intuitionistic Logics,* D Reidel, Dordrecht, 1983.

Fitting, Melvin, ''First-Order Modal Tableaux,'' *Journal of Automated Reasoning*, vol. 4, pp. 191-213, 1988.

Fitting, Melvin, ''Destructive modal resolution,'' *Journal of Logic and Computation*, vol. 1, no. 1, pp. 83-97, 1990a.

Fitting, Melvin, *First Order Logic and Automated Theorem Proving,* Springer-Verlag, Berlin, 1990b.

Frisch, Alan M, ''A general framework for sorted deduction: fundamental results on hybrid reasoning,'' in *Proceedings, KR89: Principles of Knowledge Representation and Reasoning*, ed. R J Brachman, H Levesque and R Reiter, pp. 126-136, Morgan Kaufmann, Los Altos, 1989.

Frisch, Alan M, ''The substitutional framework for sorted deduction: fundamental results on hybrid reasoning,'' *Artificial Intelligence*, vol. 49, pp. 161-198, 1991.

Frisch, Alan M and Scherl, Richard B, ''A constraint logic approach to modal deduction,'' in *Logics in AI*, ed. J van Eijck, Springer-Verlag, Berlin, 1990.

Frisch, Alan M and Scherl, Richard B, ''A general framework for modal deduction,'' in *Principles of Knowledge Representation and Reasoning: Proceedings of the Second International Conference*, ed. J A Allen, R Fikes, and E Sandewall, Morgan Kaufmann, 1991.

Gallier, J H, *Logic for Computer Science: Foundations of Automatic Theorem Proving,* John Wiley and Sons, New York, 1987.

Geissler, Christophe and Konolige, Kurt, ''A resolution method for quantified modal logics of knowledge and belief,'' in *TARK I*, ed. J Halpern, Morgan Kaufmann, 1986.

Gent, Ian P, ''A Sequent System for Lewis's counterfactual Logic VC,'' Research Report 168, Department of Computer Science, University of Warwick, 1990. Revised version to appear in *Notre Dame Journal of Formal Logic*

Gent, Ian P, ''Finding Problems in Knowledge Bases Using Modal Logics,'' Research Report 173, Department of Computer Science, University of Warwick, 1991a.

Gent, Ian P, ''Theory Tableaux,'' Technical Report 91-62, Mathematical Sciences Institute, Cornell University, Ithaca, NY , 1991b.

Gent, Ian P and Cohn, Anthony G, ''Reasoning About Hybrid Reasoning: a Discussion,'' Research Report 193, Department of Computer Science, University of Warwick, 1991. (presented at the AAAI Symposium on the Principles of Hybrid Reasoning, 1991)

Goré, Rajeev, *Cut-free Sequent and Tableau Systems for Propositional Normal Modal Logics,* PhD thesis, University of Cambridge , in preparation, 1991.

Goré, Rajeev and , Marcello D'Agostino, *A more efficient alternative to semantic tableau: theory and experiments* , in preparation, 1991.

Hailperin,Theodore, ''A theory of restricted quantification I,'' *Journal of Symbolic Logic*, vol. 22, no. 1, pp. 19-35, 1957a.

Hailperin,Theodore, ''A theory of restricted quantification II,'' *Journal of Symbolic Logic*, vol. 22, no. 2, pp. 113-129, 1957b.

Halpern, Joseph Y and Moses, Yoram, ''A guide to the modal logics of knowledge and belief: preliminary draft,'' in *Proceedings, International Joint Conference on Artificial Intelligence*, pp. 480-490, Morgan Kaufmann, 1985.

Hovsepian, Félix, *A Metalogical Analysis of Vagueness,* PhD thesis, University of Warwick, in preparation, 1992.

Hughes, G E and Cresswell, M J, *An Introduction to Modal Logic,* Methuen, London, 1968.

Hähnle, Reiner, *Towards an efficient tableau proof procedure for multiple-valued logics*, 1990. To appear in Proceedings Workshop Computer Science Logic, Heidelberg 1990

Hähnle, Reiner, ''Uniform notation of tableau rules for multiple-valued logics,'' in *Proceedings of the Twenty-first International Symposium on Multiple-Valued Logic*, pp. 238-245, IEEE Computer Society Press, Los Alamitos, California, 1991.

Jackson, Peter and Reichgelt, Han, ''A general proof method for first order modal logic,'' in *Proceedings, International Joint Conference on Artificial Intelligence*, pp. 942-944, Morgan Kaufmann, 1987.

Jackson, Peter and Reichgelt, Han, ''A general proof method for modal predicate logic without the Barcan formula,'' in *AAAI: 1988 Proceedings*, pp. 177-181, Morgan Kaufmann, 1988. (Proceedings, Conference of the American Association for Arificial Intelligence)

Jackson, Peter and Reichgelt, Han, ''A general proof method for modal predicate logic,'' in *Logic-based Knowledge Representation*, ed. P Jackson, H Reichgelt, & F van Harmelen, pp. 177-228, MIT Press, 1989. Chapter 8

Jeffrey, R C, *Formal Logic: Its Scope and Limits,* McGraw Hill, New York, second edition 1981. (cited in D'Agostino (1990))

Lloyd, J W, *Foundations of Logic Programming,* Springer-Verlag, Berlin, 1984.

Manthey, Rainer and Bry, Francois, ''SATCHMO: a theorem prover implemented in Prolog,'' in *9th International Conference on Automated Deduction*, ed. Ewing Lusk and Ross Overbeek, pp. 415-434, Springer-Verlag, Berlin, 1988.

Mates, Benson, *Elementary Logic,* Oxford University Press, New York, Second edition 1972. (cited in Boolos (1984))

McCharen, J, Overbeek, R, and Wos, L, ''Complexity and related enhancements for automated theorem-proving programs,'' *Computers and Mathematics with Applications*, vol. 2, pp. 1-16, 1976 . (cited in Wos (1988))

Mints, G, ''Gentzen-type systems and resolution rule. Part II. Predicate logic,'' in *Logic Colloquium 1990.*

Mints, G, ''Gentzen-systems and resolution rules: Part I: Propositional logic,'' in *COLOG-88*, ed. P Martin-Lof and G Mints, pp. 198-231, Springer-Verlag , Berlin, 1988.

Miura, Satoshi, ''Embedding of modal predicate logics into lower predicate calculus I,'' *International Logic Review*, no. 28 (vol. XIV, number 2), pp. 94-105, Dicembre 1983.

Mondadori, Marco, ''An improvement of Jeffrey's deductive trees,'' Discussion Paper 7, Ann. Univ. Ferrara. Sez.III. Filosofia, 1989.

Morgan, Charles G, ''Methods for automated theorem proving in nonclassical logics,'' *IEEE Transactions on Computers*, vol. C-25, no. 8, pp. 852-862, August 1976.

Morris, James B, ''E-Resolution: extension of resolution to include the equality relation,'' in *Proceedings of the International Joint Conference on Artificial Intelligence*, ed. Donald E Walker and Lewis M Norton, pp. 287-294, Distributed by William Kaufmann Inc, Los Altos, 1969.

Ohlbach, H. J. and Siekmann, J., ''The Markgraf Karl Refutation Procedure,'' SEKI Report SR-89-19, Fachbereich Informatik, Universität Kaiserslautern, 1989.

Ohlbach, Hans-Jürgen, ''Context Logic,'' SEKI Report SR-89-08, Fachberich Informatik, Univesitat Kaiserslautern, 1989.

Ohlbach, H J, ''A resolution calculus for modal logics,'' in *9th International Conference on Automated Deduction*, ed. Ewing Lusk and Ross Overbeek, pp. 500-516, Springer-Verlag, Berlin, 1988a.

Ohlbach, H J, ''A Resolution Calculus for Modal Logics,'' D386, Universität Kaiserslautern, 1988b. PhD Thesis

Ohlbach, H J, ''Semantics Based Translation Methods for Modal Logics,'' SEKI Report SR-90-11, Fachbereich Informatik, Universität Kaiserslautern, Kaiserslautern, Germany, 1990.

Paech, Barbara, ''Gentzen-systems for propositional temporal logics,'' in *CSL-88*, ed. E Borger et al, pp. 240-253, Springer-Verlag , Berlin.

Pelletier, Francis Jeffry, ''Seventy-five problems for testing automatic theorem provers,'' *Journal of Automated Reasoning*, vol. 2, pp. 191-216, 1986. Errata in Volume 4 (1988), pages 235-236

Petermann, Uwe, ''An extended Herbrand theorem for first-order theories with equality interpreted in partial algebras,'' in *Mathematical Foundations of Computer Science 1989*, ed. A Kreczmar and G Mirkowska, pp. 408-417, Springer-Verlag , Berlin, 1989.

Petermann, Uwe, ''Towards a connection procedure with built in theories,'' in *Logics in AI*, ed. J van Eijck, pp. 444-453, Springer-Verlag, Berlin, 1990a.

Petermann, Uwe, ''Building in Theories into a First-Order Proof Procedure based on the Connection Method,'' NTZ-Nr. 16/90, Naturwissenschaftlich-Theoretisches Zentrum, Leipzig University , 1990b.

Petermann, Uwe, ''Building in equational theories into the connection method,'' in *Fundamentals of Artificial Intelliogence Research*, ed. Ph Jorrand and J Kelemen, pp. 156-169, Springer-Verlag, Berlin, 1991.

Prawitz, Dag, ''An improved proof procedure,'' *Theoria*, vol. 26, pp. 102-139, 1960.

Prawitz, Dag, ''Advances and problems in mechanical proof procedures,'' in *Machine Intelligence 4*, ed. B Meltzer & D Michie, pp. 59-71, Edinburgh University Press, 1969.

Pym, David and Wallen, Lincoln, ''Investigations into proof search in a system of first order dependent function types,'' in *10th International Conference on Automated Deduction*, ed. M E Stickel, pp. 236-250, Springer-Verlag, Berlin, 1989.

Robinson, J A, ''A machine oriented logic based on the resolution principle,'' *Journal of the Association for Computing Machinery*, vol. 12, pp. 23-41, 1965.

Scherl, Richard B, *Hybrid reasoning and modal logic theorem proving*, 1991. (presented at the AAAI Symposium on the Principles of Hybrid Reasoning, 1991)

Schmidt-Schauβ, Manfred, *Computational Aspects of an Order-Sorted Logic with Term Declarations,* Springer-Verlag, Berlin, 1989.

Schmitt, P H and Wernecke, W, ''Tableau calculus for order sorted logic,'' in *Sorts and Types in Artificial Intelligence*, ed. K H Bläsius, U Hedtstück, C-R Rollinger, pp. 49-60, Springer-Verlag , Berlin, 1989.

Siekmann, Jörg H, ''Unification Theory,'' in *ECAI 86 Proceedings*, vol. 2, pp. vi-xxxv , 1986. (The 7th European Conference on Artificial Intelligence)

Simmons, Harold, *Sally Popkorn's Modular Modal Manual: An introduction to modal logic*, In preparation, 1992.

Smullyan, Raymond M, *First Order Logic,* Springer-Verlag, Berlin, 1968.

Smullyan, Raymond M, ''Abstract Quantification Theory,'' in *Intuitionism and Proof Theory*, ed. A Kino, J Myhill and R E Vesley, pp. 79-91, North-Holland, Amsterdam, 1970.

Stickel, Mark E, ''Automated Deduction by Theory Resolution,'' *Journal of Automated Reasoning*, vol. 1, pp. 333-355, 1985.

Stickel, M E, ''Schubert's Steamroller Problem: Formulations and Solutions,'' *Journal of Automated Reasoning*, vol. 2, pp. 85-101, 1986.

Swart, H C M De, ''A Gentzen- or Beth-Type System, a practical decision procedure and a constructive completeness proof for the counterfactual logics VC and VCS,'' *Journal of Symbolic Logic*, vol. 48, no. 1, pp. 1-20, 1983.

Torsun, I S and Manning, K J, ''Application of Temporal Logic to Tax Regulations,'' CS-45-90, Department of Computing, University of Bradford, United Kingdom, 1990.

Wallen, Lincoln A, ''Generating connection calculi from tableau and sequent based proof systems,'' in *Artificial Intelligence and its Applications*, ed. A G Cohn & J R Thomas, pp. 35-50, John Wiley & Sons, 1986.

Wallen, Lincoln A, *Automated Proof Search in Non-Classical Logics: Efficient Matrix Proof Methods for Modal and Intuitionistic Logics,* University of Edinburgh, Edinburgh, 1987a. (PhD Thesis)

Wallen, Lincoln A, ''Matrix proof methods for modal logics,'' in *Proceedings, International Joint Conference on Artificial Intelligence*, pp. 917-923, Morgan Kaufmann, 1987b.

Wallen, Lincoln A, *Automated Proof Search in Non-classical Logics,* MIT Press, Cambridge, Massachusetts, 1989. *(a revised version of his PhD thesis, above)*

Wallen, Lincoln A and Wilson, Gregory V, ''A computationally efficient proof system for S5 modal logic,'' in *Advances in Artificial Intelligence*, ed. C Mellish & J Hallam, pp. 141-153, Wiley and Sons, 1987.

Wirsing, Martin, ''Algebraic Specification,'' in *Handbook of Theoretical Computer Science, Volume B: Formal Models and Semantics*, ed. Jan Van Leeuwen, pp. 675-788, Elsevier, Amsterdam, 1990. (also issued by MIT Press, Cambridge, MA)

Wos, Larry, *Automated Reasoning: 33 Basic Research Problems,* Prentice Hall, Englewood Cliffs, New Jersey, 1988.

Wos, Lawrence, Robinson, George A, Carson, Daniel F, and Shalla, Leon, ''The concept of demodulation in theorem proving,'' *Journal of the Association for Computing Machinery*, vol. 14, no. 4, pp. 698-709, October 1967.

Name Index

This index contains all references to the literature and to personal communications contained in the main text and in Appendix 1. Omitted are references to people in my acknowledgements as well the quotations by Henry Kautz, Peter Schickele, and Mary-Claire van Leunen. Also omitted are people whose names are used solely to refer to something else, as for example in ''Herbrand models'', ''Kripke semantics'', and ''Horn clauses.''

Index of Symbols and Notation

Below I give page references for non-standard symbols and notation used in this thesis. Sometimes two definitions are listed for a symbol, for example if its syntax and semantics are defined separately. The symbols ε and → take different meanings in different context, and so I give two entries for those.

I use the standard logical symbols **t** (truth), **f** (falsity), ∧ (conjunction), ∨ (disjunction), ⊃ (implication), ¬ (negation), ∀ (universal quantifier), and ∃ (existential quantifier). The syntax and semantics for these symbols are defined in Definitions 2.2.1 and 2.3.2.

To mark the end of proofs I use the symbol □

Below, a page number in roman type indicates that the notation is introduced in the main body of the text on that page. A page number in **bold** type indicates that the notation is used in a definition on that page. A page number in *italic* type indicates that the notation is explained in an italicised paragraph headed "Notation" on that page.

Symbol	Notes	Page(s)	Definition(s)
$\forall x_{:\rho(x)}$	universal restricted quantification	1, **11**, **12**	2.2.1, 2.3.2
$\exists x_{:\rho(x)}$	existential restricted quantification	1, **11**, **12**	2.2.1, 2.3.2
□	modal logic "box"	1, 15	
◇	modal logic "diamond"	1, 15	
$\mathcal{R}$	modal accessibility relation	1, 15	
→	(in clauses)	**10**	2.1.1, 2.1.2
→	(in sort function declarations)	17	
Σ	restriction theory	9, **10**	2.1.3
$\Sigma\mathcal{L}$	logic of restricted quantification	9, 10	
$\mathcal{L}$	parts of $\Sigma\mathcal{L}$ distinct from Σ	10	
ε	distinguished name	10	
ε	distinguished world	15	
< ··· , ··· >	model	**12**,	2.3.1, 2.3.2
$(\ )^{\mathbf{I}}$	term assignment	**12**	2.3.1
$(\ )^{\mathbf{A}}$	variable assignment	**12**	2.3.1
$(\ )^{\mathbf{I,A}}$	general semantic assignment	**12**	2.3.1, 2.3.2
$\models_{\Sigma}$	entailment within Σ	**13**	2.3.3
⊩	forcing	63	
0	negative polarity	**13**	2.4.1, 2.4.2
1	positive polarity	**13**	2.4.1, 2.4.2
α	conjunctive formula or position	**14**, *30*	2.4.4
α_1, α_2	subformulas or subpositions of α	**14**, *30*	2.4.4
β	disjunctive formula or position	**14**, *30*	2.4.5
β_1, β_2	subformulas or subpositions of β	**14**, *30*	2.4.5
γ	universal formula or position	**14**, *32*	2.4.6
γ_0, γ_r	subformulas or subpositions of γ	**14**, *32*	2.4.6
δ	existential formula or position	**15**, *32*	2.4.7
δ_0, δ_r	subformulas or subpositions of δ	**15**, *32*	2.4.7
τ_m	propositional modal translation	15	
next	skolem function for serial modal logics	15	
τ_m^c	constant domain modal translation	16	
τ_m^v	varying domain modal translation	16	
⊐	subsort relation	17	
$\mathcal{T}$	expansion tree	20	
⊗	closed branch in expansion tree	*22*	

$(\ )^c$	complement (other than τ_m^c, above)	*22*	
Occ	set of occurrences	29	
$\sim_{\mathcal{T}}$	equivalence relation on **Occ**	**30**	3.5.1
$\ll$	subposition	**30**	3.5.2
P	set of positions	**30**	3.5.2
$\mathbf{P}^{\alpha}$, $\mathbf{P}^{\alpha_1}$, etc.	set of positions of given type	*30*, *32*	
lab( )	label of a position	30	
pol( )	polarity of a position	30	
(*1*)	side condition	**32**	4.1.1
(*2*)	side condition	**32**	4.1.1
(*3*)	side condition	**32**	4.1.1
(*4*)	side condition	**32**	4.1.1
(*5*)	side condition	**43**	5.1.2
(*6*)	side condition	**83**	9.1.1
$\mathbf{M}_{\lambda}$	model for a branch λ	**39**	4.4.2.1
LHM_{Σ}	(Least Herbrand Model)	**43**	5.1.1
new( )	new name in a Σ-atom	47	5.2.2
$\rho(\cdots;\cdots)$	distinction of eligible argument	*49*	
$(\ )\sigma$	assignment	*50*	
$[\cdots := \cdots, \cdots]$	assignment	*50*	
$\hat{\cdot}$	indication of marking	*60*	
$\sqsubset$	binary relation on position	**67**	7.1.6
$\lhd$	binary relation on positions	*67*, **68**	7.1.7
$X[A]$	formula with distinguished subformula	*75*	
$\mathcal{Q}$	generic quantifier	*77*	

Index of Expansion Systems and Expansion Rules

To summarise some naming conventions for expansion systems:

- If **S** is the name of a propositional expansion system, then Σ**S**, Σ**InstS**, and Σ**branchS** refer to three different expansion systems for restricted quantification using **S**'s propositional rules.
- Names of expansion systems decorated with a list of numbers, for example Σ**S**(*1345*) refer to the expansion system for restricted quantification, in which the listed side conditions are demanded.

Propositional Expansion Systems

The following table lists propositional expansion systems used in this thesis, with the names of rules used in that system and the section of the thesis in which the system is introduced.

Name	Rules	Section
Tab	(α), (β)	3.1.2
Tab with equivalence	**Tab** + (**1**≡), (**0**≡)	3.1.4
KE	(A), (B1), (B2), (PB)	3.1.3
KE with equivalence	**KE** + (**1**E+L), (**1**E-L), (**0**E+L), (**0**E-L), etc.	3.1.5
analytic **KE**	(A), (B1), (PBβ_2) *or* (A), (B2), (PBβ_1)	3.3 (Theorem 3.3.8)

Extensions of Propositional Systems for Restricted Quantification

If **S** is a propositional expansion system, then it can be extended for restricted quantification in one of the following ways.

Name	Rules	Relevant Side Conditions	Section
Σ**S**	**S** + (γ), (δ)	(*1*), (*2*), (*3*), (*4*), (*5*)	4.1
Σ**InstS**	**S** + (γ-Inst), (δ-Inst)	(*1*), (*3*), (*4*), (*5*)	8.1
Σ**branchS**	**S** + (γ-branch), (δ)	(*1*), (*3*), (*4*), (*6*)	9.1

The definitions of the side conditions (*1*) to (*6*) are indexed in the Index of Symbols and Notation.

List of Definitions

List of Theorems and Corollaries

Appendix 1

The Use of Sequences as Prefixes for Modal Logics

One feature of much work in modal logic theorem proving is the use of *sequences* as prefixes, as for example in the work of Fitting (1983), Wallen (1989), and Frisch and Scherl (1990, 1991). In this thesis, however, I used simple names for prefixes (prefixes to the forcing relation $\Vdash$, that is) as in the examples in Section 4.2. In this appendix I show that it is not difficult to derive proof systems using sequences as prefixes from the work presented in this thesis.

This appendix provides a proof system very close to Fitting's prefixed tableaux (1983, Chapter 8). Also, it would be a very good starting point for the derivation of matrix-based characterisations of validity for modal logics which would be very similar to Wallen's (1989). (The matrix characterisations actually presented in Chapter 7 are further from Wallen's precisely because sequences are not used.) The arguments from Chapter 7 could be repeated for the proof system presented here very easily. The arguments presented here do not have the generality of Frisch and Scherl's (1990, 1991) or their completeness. Here I concentrate on modal logics, but the arguments seem amenable to generalisation without too much difficulty.

Where the modal accessibility theory satisfies alphabetical monotonicity or non-vacuity, one can define a sequence based proof system as follows. As in the rest of this thesis, I consider only *ground*-based proof systems and I do not consider unification.

Let us say that a *sequence* is any finite sequence of positive integers, including the empty sequence ε. *I will refer to a typical sequence as* p *and write* $p,2,1$ *to refer to the sequence* p *followed by* 2 *followed by* 1. *If* p *and* q *are sequences I will write* pq *for their concatenation.*

We build an accessibility relation R_0 on the set of prefixes by demanding that for any sequence p, and for any integer i, that

$$p\, R_0\, p,i\ .$$

Now, the conditions of reflexivity, symmetry, and transitivity translate into simple additional conditions on this relation as follows:

Property	Additional Condition on R_0
reflexive	$p\, R_0\, p$ for any sequence p
symmetry	$p,i\, R_0\, p$ for any sequence p, integer i
transitive	$p\, R_0\, pq$ for any sequence p, non-empty sequence q

The great advantage of sequence-based proof systems for modal logics is that reasoning within the modal theory is replaced with reasoning about the single, fixed, binary relation R_0. This opens the way for the use of sequence unfication in automated theorem proving. (The condition of seriality is dealt with differently, as will be discussed below.)

In Section 4.2, I briefly discussed the fact that one can easily derive modal logic expansion rules from the rules (γ) and (δ). They would be as follows (I introduce only the rules for positive $\Box$ formulas and positive $\Diamond$ formulas, the rules for negative formulas being obvious once these are given).

Modal Logic Expansion Rules using Names

(**1**□)	$\dfrac{w \Vdash \mathbf{1}\Box\phi}{w_1 \Vdash \mathbf{1}\phi}$	(**1**◇)	$\dfrac{w \Vdash \mathbf{1}\Diamond\phi}{w_1 \Vdash \mathbf{1}\phi \quad w \mathcal{R} w_1}$

The side condition (*1*) now demands of an application of (**1**□) that we can prove $w \mathcal{R} w_1$ and the side condition (*2*) now demands of an application of (**1**◇) that w_1 is a new name on the branch.

Although previously we have regarded names as being devoid of internal structure, we can equally well now look on the set of sequences as a set of names. Note that, as defined above, there are only countably many sequences. The next point is to observe that the rule **1**◇ introduces "$w \mathcal{R} w_1$". If names are sequences then we can always choose w_1 to be w,i for some integer i. In that case we always have $w R_0 w_1$. Doing this consistently would ensure that $\mathcal{R}$ is always subsumed by R_0.

Having shown that $\mathcal{R}$ is subsumed by R_0 the next step is to show that we can replace the use $\mathcal{R}$ in side condition (*2*) by R_0. Fortunately the work in this thesis makes this rather easy.

The crucial point is to observe that, barring only the modal logic thoery containing only symmetry and transitivity, all modal logic theories satisfy alphabetical monotonicity or non-vacuity (as shown in Chapter 6). Where alphabetical monotonicity holds, we can replace the side condition (*2*) by (*5*), namely:

> "provided $w \mathcal{R} w_1 \in LHM_{\Sigma}(\mathcal{T})$"
>
> where $LHM_{\Sigma}(\mathcal{T})$ is the least Herbrand model of all literals $w \mathcal{R} w_1$ introduced by the rule (**1**◇) *anywhere* in the tree $\mathcal{T}$.

Except in the case of seriality, it is clear that R_0 represents precisely the least Herbrand model in the appropriate theory of the complete set of atoms

$$\{\, p \mathcal{R} p,i \mid p \text{ any sequence, } i \text{ any integer} \,\}.$$

For any given finite subset of this set of atoms, we can construct a tree which contains that subset, and moreover using a formula which is equivalent to any given formula! This is in fact easy. Say we have a formula $\mathbf{1}X$ and we wish to include the atoms

$$\{\, \varepsilon \mathcal{R} 3 \,,\, 2 \mathcal{R} 2,1 \,,\, 1,3 \mathcal{R} 1,3,4 \,\}.$$

Consider the following formula:

$$\varepsilon \Vdash \mathbf{1}\, X \vee (\Diamond \mathbf{f}) \vee (\Diamond \mathbf{f} \wedge \Box\Diamond \mathbf{f}) \vee (\Diamond \mathbf{f} \wedge \Box\Diamond \mathbf{f} \wedge \Box\Box\Diamond \mathbf{f})$$

The recurrent use of ◇**f** means that it is equivalent to X in any of the modal logics under consideration here.

We could start a tree for this formula with an initial four branches for the first four formulas. On the second branch (for ◇**f**) we could use the rule (**1**◇) to introduce $\varepsilon \mathcal{R} 3$. On the third branch (for ◇**f**∧□◇**f**) we could introduce first $\varepsilon \mathcal{R} 2$ and then $2 \mathcal{R} 2,1$. On the fourth branch (for ◇**f**∧□◇**f**∧□□◇**f**) we could introduce $\varepsilon \mathcal{R} 1$ followed by $1 \mathcal{R} 1,3$, and finally $1,3 \mathcal{R} 1,3,4$. This technique can be generalised to introduce an arbitrary finite set of atoms.

Given the argument above, in an expansion tree for any formula X it is quite sound to assume the presence of any arbitary finite set of atoms. Because of this, and since all the modal theories under consideration are compact (as are all theories considered in this thesis) we can safely assume the presence of the entire set $\{\, p \mathcal{R} p,i \mid p \text{ any sequence, } i \text{ any integer} \,\}$. As mentioned above, R_0 is the least Herbrand model of this set, so the condition (*5*) is equivalent to

> "provided that $w \mathcal{R} w_1 \in R_0$."

One point remains to be resolved. This is the distinction between serial logics such as **D** and non-serial logics such as **K**. Again, the work in this thesis provides a simple solution. Recall from Chapter 6 that

logics in which neither seriality or reflexivity hold satisfy alphabetical monotonicity but not non-vacuity. Then recall from Chapter 5 that in this case we must demand the condition (*3*), that only names (sequences, now) already appearing on the branch may be used in applications of (γ) (or (**1**□) here). For example suppose $p \Vdash \mathbf{1}\Box\mathbf{f}$ appeared in a branch. Although of course for any i we have $p\,R_0\,p,i$, this fact cannot be used unless for some i the sequence p,i has been introduced on the branch by the rule (**1**◇). This is all that is needed to separate the non-serial case from the serial or reflexive case.

In summary, the various arguments above show the soundness and completeness of the following proof system for all modal logics considered in this thesis, excepting the modal logic for which the accessibility relation is only symmetry and transitivity. In the statements below, the relation R_0 refers to the relation defined for the appropriate modal theory.

Modal Logic Expansion Rules using Sequences

(**1**□)	$\dfrac{p \Vdash \mathbf{1}\Box\phi}{q \Vdash \mathbf{1}\phi}$	(**1**◇)	$\dfrac{p \Vdash \mathbf{1}\Diamond\phi}{p,i \Vdash \mathbf{1}\phi}$

where applications of (**1**□) must satisfy

provided that $p\,R_0\,q$;

and applications of (**1**◇) must satisfy

provided that i is an integer and p,i is a new sequence on the branch;

and additionally, if the modal accessibility theory is neither serial nor reflexive, applications of (**1**□) must satisfy

q is a sequence that has been introduced by some (**1**◇) expansion on the branch.