Computations
in
Propositional Logic

ILLC Dissertation Series 1996-1

institute *for* logic, language *and* computation

For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam
phone: +31-20-5256090
fax: +31-20-5255101
e-mail: illc@fwi.uva.nl

Computations in Propositional Logic

Academisch Proefschrift

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam,
op gezag van de Rector Magnificus
prof.dr P.W.M. de Meijer
ten overstaan van een door het college van dekanen ingestelde
commissie in het openbaar te verdedigen in de
Aula der Universiteit
op dinsdag 12 maart 1996 te 15.00 uur

door

Alex Hendriks

geboren te Deventer.

Promotor: Prof.dr. G.R. Renardel de Lavalette
Faculteit der Wiskunde en Natuurwetenschappen
Rijksuniversiteit Groningen
Blauwborgje 3
9747 AC Groningen

Co-promotor: Dr. D.H.J. de Jongh
Faculteit der Wiskunde, Informatica, Natuurkunde en Sterrenkunde
Universiteit van Amsterdam
Plantage Muidergracht 24
1018 TV Amsterdam

The investigations were supported by the Mathematical Research Foundation (SMC), which is subsidized by the Netherlands Organization for Scientific Research (NWO).

CIP-GEGEVENS KONINKLIJKE BIBLIOTHEEK, DEN HAAG

Hendriks, Alex

Computations in propositional logic / Alex Hendriks. -
Amsterdam : Institute for Logic, Language and
Computation, Universiteit van Amsterdam. - Ill. - (ILLC
dissertation series ; 1996-01)
Proefschrift Universiteit van Amsterdam- - Met index, lit.
opg. - Met samenvatting in het Nederlands.
ISBN 90-74795-44-7
NUGI 855
Trefw.: propositielogica / intuïtionisme / modale logica.

Contents

Dankwoord 1

1 General Introduction 3
- 1.1 Outline of the thesis . . . 7
- 1.2 General preliminaries . . . 8

2 Semantic Types and Exact Models 15
- 2.1 Introduction . . . 15
- 2.2 Preliminaries . . . 18
- 2.3 Types in **CpL** . . . 21
- 2.4 Types in modal logic . . . 24
- 2.5 Types and reductions in **IpL** . . . 31
- 2.6 Calculations in exact models . . . 37
- 2.7 Games and bisimulations . . . 39

3 Exact Models in IpL 41
- 3.1 Introduction . . . 41
- 3.2 Preliminaries . . . 43
- 3.3 The $[\wedge, \vee]$ fragments . . . 45
  - 3.3.1 $[\wedge]$ and $[\vee]$ fragments . . . 48
- 3.4 The $[\wedge, \vee, \neg]$ fragments . . . 48
  - 3.4.1 The $[\wedge, \neg]$ fragments . . . 52
  - 3.4.2 The $[\wedge, \neg\neg]$ fragments . . . 57
  - 3.4.3 The $[\wedge, \vee, \neg\neg]$ fragments . . . 59
  - 3.4.4 The $[\vee, \neg]$ fragments . . . 62
  - 3.4.5 The $[\vee, \neg\neg]$ fragments . . . 65
- 3.5 The $[\wedge, \to, \neg]$ fragments . . . 69
  - 3.5.1 The $[\to, \neg]$ fragments . . . 77
  - 3.5.2 The $[\wedge, \to, \neg\neg]$ fragments . . . 79
  - 3.5.3 The $[\to, \neg\neg]$ fragments . . . 83

3.6 The $[\wedge, \to]$ fragments . . . 85
3.6.1 The $[\to]$ fragments . . . 90

4 Restricted nesting of implication in IpL 93
4.1 Introduction . . . 93
4.2 Preliminaries . . . 93
4.3 Semantic types in $\mathbf{IpL}^n_m$ . . . 94
4.4 The n, m-types in **IpL** . . . 96

5 Exactly provable L formulas 99
5.1 Introduction . . . 99
5.2 Preliminaries . . . 100
5.3 Exactly provable formulas in $\mathbf{L}^n$ . . . 102
5.4 Maximal exactly provable formulas . . . 104
5.5 Calculating exactly provable formulas . . . 108

6 A family of propositional testers 113
6.1 Introduction . . . 113
6.2 Preliminaries . . . 113
6.3 CpLtest: a **CpL** tester . . . 114
6.4 IpLtest: an **IpL** tester . . . 118
6.5 Ktest: a tester for **K** . . . 123
6.6 Other testers for modal propositional logic . . . 129
6.6.1 Ttest: a **T** tester . . . 129
6.6.2 K4test: a **K4** tester . . . 129
6.6.3 S4test: an **S4** tester . . . 134
6.6.4 Ltest: an **L** tester . . . 134

A Computer programs 137
A.1 Preliminaries . . . 137
A.2 The mkDiag program . . . 138
A.3 A simple **CpL** tester . . . 142
A.4 The IpLtest program . . . 144
A.5 Testers for modal logic . . . 146

B Output of computer programs 153
B.1 The diagram of the **IpL** fragment $[\to, \neg]^2$ . . . 153
B.2 The diagram of $\mathbf{H}^2_3$ . . . 163
B.3 The diagram of the fragment $\mathbf{IpL}^2_1$ . . . 166
B.4 The exactly provable formulas in $\mathbf{L}^1_1$ . . . 168

C Table of fragments in IpL 171

Bibliography **173**

List of symbols **177**

Index **179**

Samenvatting **183**

Dankwoord

Graag zou ik hier iedereen, met naam en toenaam, bedanken die op de een of andere manier heeft bijgedragen aan het totstandkomen van dit proefschrift. Iets wat helaas niet gaat. Maar temidden van de velen in wie ik mijn leermeesters, voorbeelden en toeverlaten herken, zijn er bij deze gelegenheid enkele die beslist niet ongenoemd kunnen blijven.

Allereerst wil ik hier uitdrukkelijk mijn promotor Gerard Renardel de Lavalette en mijn co-promotor Dick de Jongh bedanken. Zonder hun aanmoediging was ik er nooit aan begonnen, zonder hun steun was het nooit afgekomen, zonder hun kritiek was het nooit wat geworden en zonder hun vriendschap was het allemaal ook minder de moeite waard geweest.

Dick heeft mij bovendien de afgelopen vier jaar als medebewoner op zijn kamer geduld. Een privilege waaraan de overige bewoners op de gang in de loop der tijd nog meer glans wisten te geven. Ik dank hierbij met name professor Troelstra, Andreja Prijatelj en Paul van Ulsen voor zowel de stimulerende als de amusante gesprekken die ik met hen mocht voeren.

De vakgroep toegepaste logica van de faculteit voor Wijsbegeerte van de Rijksuniversiteit van Utrecht ben ik erkentelijk voor het jaar (1991) dat ik in Utrecht heb mogen werken om de grondslag te leggen voor dit proefschrift.

Met Henk van Riemsdijk, John Tromp en Jan Zwanenburg heb ik de afgelopen jaren korte of langere tijd mogen samenwerken en het enthousiasme kunnen delen over de mogelijkheden die computers bieden voor onderzoek in de logica.

Volodya Shavrukov wil ik hier nogmaals bedanken voor zijn kritiek waarmee hij mij behoed heeft voor een ernstig misverstand over de structuur van de exacte modellen in de modale logica.

De leden van mijn promotiecommissie, professor Van Benthem, professor Kamp, dr. Rodenburg, professor Troelstra en dr. Visser, ben ik zeer erkentelijk voor hun commentaar en suggesties voor verbeteringen.

Dankbaar ben ik verder voor het geduld en de welwillendheid waarmee vrienden, familieleden en mijn collega's bij het GAK de afgelopen jaren mijn verhalen over exotische propositielogica's hebben aangehoord.

Dankbaar ben ik ook mijn schoonouders voor hun vertrouwen en steun waardoor ik mijn studie heb kunnen afmaken.

Meer nog dan al deze dank verdienen mijn vrouw Hannie, mijn dochters Maartje en Aletta en onze pleegzoons van de afgelopen jaren, Rob en Gabor. Zij hebben niet alleen het geduld opgebracht om mij aan het werk te laten, zij hebben mij er ook dikwijls uit losgerukt om samen met hen nieuwe energie op te doen.

Dit proefschrift draag ik op aan mijn ouders, Jan Hendriks en Aleida Hendriks-Velders. Hun betrokkenheid was mij ook bij het totstandkomen van dit proefschrift tot steun. Ik ben blij dat ik mijn vader het resultaat heb kunnen laten zien.

Amsterdam Lex Hendriks
Januari, 1996.

Chapter 1

General Introduction

The main topic of this thesis is the representation of fragments of intuitionistic and modal propositional logic by (usually finite) structures, called *exact models*. One of the reasons for the interest in properties of fragments with such a finite representation is the possibility of designing computer programs to decide derivability within the fragment. In general, this kind of program, based on checking the validity of formulas in a model, is much more efficient than traditional theorem proving. This efficiency of 'model checking versus theorem proving' has in recent years attracted the attention of researchers in artificial intelligence and knowledge representation [HV 91].

For the benefits of model checking we have to pay a price. Theorem provers are 'general purpose' tools, accepting any formula in the language of the logic (obviously with certain practical limitations). However, finite representations such as exact models exist only when we cut down the expressive power of the language.

In this thesis we focus our attention on *finite fragments* of propositional logics, languages with restrictions on the use of atomic subformulas and connectives, that have a finite *Lindenbaum algebra* or *diagram* as we prefer to call it here.

The structure of these finite diagrams can be calculated and studied using efficient computer programs based on model checking. Knowledge of the structure of diagrams can be used in constructing new finite complete models.

The history of this kind of research into the structure of finite fragments can be traced back to the calculation of diagrams by Skolem [Skolem 13] and Lindenbaum's suggestion to use (equivalence classes of) formulas in semantics [Mostowski 65]. The discovery of the lattice of the one-variable fragment of intuitionistic propositional logic by Rieger [Rieger 49] and its rediscovery by Nishimura [Nishimura 60] proves that, although perhaps not always very prominent, the interest in the subject remained in the years after.

After the introduction of semantic tableaux by Beth [Beth 55], Kanger and Hintikka, and the invention of Kripke semantics for modal as well as intuitionistic logic by Kripke [Kripke 65] and others, a more systematic investigation of the semantical fine structure of fragments seemed possible.

As Beth pointed out in [Beth 55], the strongly mechanical character of his semantic tableau procedures suggests the possibility of constructing a *logical machine*. In 1955 Beth imagined this futuristic 'computer' to display its results using a crossbreed of a traffic light and a telegraph. A *red* light would announce a proof, to be produced on a strip of paper, and a *yellow* light would announce the machine to print a finite counter-example.

The construction of the logical machine would of course depend on the logic used. Only in those cases where the derivability of a formula from a finite set of formulas is decidable, one may expect the machine always to halt after a finite amount of time.

In the case of predicate logic, adding a green light to the machine announcing the construction of an infinite tableau would make the implementation of the specifications impossible (if on every input the machine has to switch on one of the lights after a finite period of time).

Nowadays, at a time where there are probably more computers than traffic lights around, it is no problem to implement Beth's logical machines as computer programs. Of course, if we want the machine to halt on every input, such a computer program is only possible for decidable logics, such as most propositional logics.

In the early sixties, as soon as computers came within the reach of university scientists, Beth stimulated his students De Jongh and Kamp to develop computer programs deciding derivability and compute diagrams in intuitionistic propositional logic (**IpL**).

In 1963 De Jongh and Kamp succeeded in making the computer decide correctly whether a formula was derivable in **IpL**. If not, the program produced the description of a Kripke model that served as a counter-example.

However, the program was too time-consuming to be of any practical use in studying diagrams. In the late seventies, this line of research was picked up by the author [Hendriks 80] who wrote Algol68 programs that could decide derivability in **IpL** and compute small diagrams. Improved results were obtained by Van Riemsdijk [Riemsdijk 85] with Pascal programs.

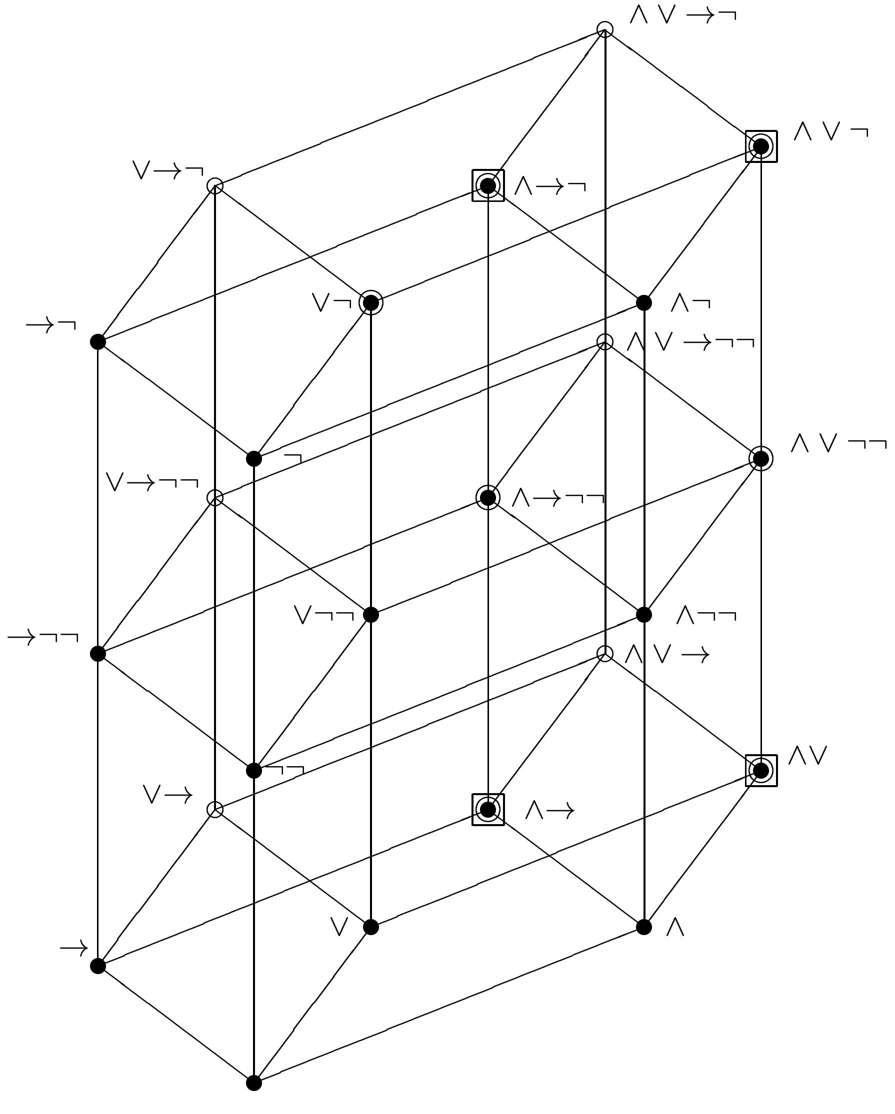
These computer programs, using algorithms based on the semantic tableaux method, realized the kind of logical machine that Beth envisaged 25 years earlier.

By that time the history of the subject had also made a detour in algebra. Investigating the algebraic structure of diagrams of $[\rightarrow]$ fragments of **IpL**, Diego proved that all diagrams of $[\rightarrow]$ with a finite number of propositional variables are finite [Diego 66].

Independently, Urquhart gave a simpler prove in [Urquhart 74] and in 1975, De Bruijn proved the same result for all diagrams of $[\wedge, \rightarrow]$ [Bruijn 75a]. In this proof De Bruijn introduced the notion of an *exact model* . An exact model is a part of the Lindenbaum algebra, such that the lattice of upward closed subsets of the exact model (ordered by inclusion) is isomorphic to the entire Lindenbaum algebra.

Let us write $[\wedge, \rightarrow]^n$ for the **IpL** fragment with formulas generated from the atomic formulas $\{p_1, \ldots, p_n\}$ and the connectives $\wedge$ and $\rightarrow$. To construct the exact model of the fragment $[\wedge, \rightarrow]^3$, De Bruijn used a computer. He also published a computer program that used this exact model in deciding derivability in the fragment [Bruijn 75b].

In 1987 De Jongh, Renardel de Lavalette and the author started working on computer aided research into the structure of diagrams of **IpL**. With help from Van Riemsdijk and Tromp new computer programs were developed based on exact models. The present work reflects the results of the research that started in this group.



1. FIGURE. *The lattice of fragments in* **IpL**.

De Bruijn's concept of exact model was reformulated in the more familiar context of Kripke models, simplified and generalized to compute exact models, not only of $[\wedge, \to, \neg]$ fragments (see [JHR 91]) but also those of other fragments of **IpL** [Hendriks 93]. As a result we are now able to construct a finite complete Kripke model for every finite fragment in the lattice of fragments depicted in figure 1. Each node in this diagram stands for a certain set of connectives and hence for an infinity of fragments, one for each number of atoms. Note that the double negation ($\neg\neg$) is treated as a primitive operator.

In figure 1 the fragments with an infinite diagram (at least for more than one propositional variable) are denoted by an open circle, the others by a closed circle. Whenever the fragment (again over a finite set of atoms) has an exact model, the closed circle of the fragment is surrounded by an open circle. Fragments with an exact Kripke model are marked by a square.

As can be seen from the lattice, every fragment considered here with a finite diagram is a subfragment of a fragment with an exact Kripke model.

The fragments in the lattice above are obtained by simply deleting one or more of the usual connectives (on top of the restriction to a finite set of atoms). For several reasons, restricting the use of connectives in a more sophisticated way is an interesting alternative. Observe for example that the interplay of implication and disjunction cannot be studied in finite fragments, since in every finite fragment either one of them will be absent. Nor is there a non-trivial sequence of these 'simple' finite fragments which has $\mathbf{IpL}^n$ as its union.

If we turn to modal logic, the situation is even worse. By simply deleting connectives we will not, in general, obtain finite fragments if the modal operators are still available, and without them we are left with fragments of classical propositional logic, **CpL**.

In modal logic, there are some well-known results concerning formulas with a restricted *modal depth*, as the nesting of modal operators is usually called (see in particular [Fine 74]). In the context of an attempt to characterize formulas in provability logic using sets of worlds of a certain type in a Kripke model, these results inspired the introduction of the notion of *semantic type*. The notion of semantic type turned out to be a versatile tool also in intuitionistic propositional logic.

Just as the restriction of modal depth in modal logic, the restriction of nesting of implications in **IpL** fragments with a finite number of propositional variables yields fragments with finite diagrams that have exact Kripke models. The structure of these exact models will be studied[1] in Chapter 4. Intuitionistic propositional logic can be obtained as the limit of a sequence of fragments with an increasing nesting of implication and an increasing number of propositional variables.

The problem in provability logic, **L**, that brought us on the trail of the semantic types was the computation of the *exactly provable* formulas[2] in the fragment $\mathbf{L}_1^1$ (see Chapter 5). According to Solovay's theorem on provability interpretations for formulas of **L** [Solovay 76], the theorems of **L** are those modal formulas that are provable in Peano arithmetic (**PA**) under arbitrary arithmetical interpretations (interpreting $\Box$ as the formalized provability predicate in **PA**). If we fix the arithmetical interpretation of one or more of the propositional variables, the interpreted formulas true in

[1] Some of the research was done in cooperation with Zwanenburg [Zwanenburg 94].
[2] The term 'exact' in 'exactly provable' has no relation to its use in 'exact models'.

PA form an *interpretable theory*:

$$\{\phi(p_1, \ldots, p_n) \mid \vdash_{\mathbf{PA}} \phi^*(A_1, \ldots, A_n)\}$$

for certain arithmetic sentences $A_1, \ldots, A_n$.

There are only finitely many interpretable theories in the fragment $\mathbf{L}^n_m$, with n atoms and a nesting of the provability operator less or equal m.

If we introduce the relation $\phi \mathrel{\vdash\!\!\cdot} \psi$ for $\phi \wedge \Box\phi \vdash \psi$, then we can reformulate the condition, found by V. Shavrukov, that is both necessary and sufficient for a ϕ in $\mathbf{L}$ to be the axiom of an interpretable theory [Shavrukov 93]:

$$\text{for all } \psi, \chi \quad \phi \mathrel{\vdash\!\!\cdot} \Box\psi \vee \Box\chi \quad \Rightarrow \quad \phi \mathrel{\vdash\!\!\cdot} \psi \text{ or } \phi \mathrel{\vdash\!\!\cdot} \chi.$$

A ϕ which is the axiom of an interpretable theory in the sense that there is an interpretation such that:

$$\phi \mathrel{\vdash\!\!\cdot} \psi \quad \Leftrightarrow \quad \vdash_{PA} \psi^*$$

is called an *exactly provable* formula. The strong disjunction property above turns out to have a characterization in semantical terms by means of which it is possible to calculate the exactly provable formulas in the fragment $\mathbf{L}^1_1$.

The results based on these ideas were first published in [HJ 96].

1.1 Outline of the thesis

Each chapter starts with a short introduction and a preliminary section. The preliminaries common to all chapters can be found in section 1.2.

Chapter 2 is an introduction into the theory of semantic types and exact models. Some related results, as from [Fine 74], [Jankov 68] and [De Jongh 70], are presented in this framework.

Chapter 3 is an overview of finite fragments of **IpL** with a restricted set of connectives. For fragments with an exact model the construction of the exact model is given. Part of these results were published in [JHR 91] and most of them can also be found in [Hendriks 93]. In this chapter these results are presented for the first time within the framework of semantic types introduced in the previous chapter.

Chapter 4 deals with the structure of exact Kripke models for fragments of **IpL** with restricted nesting of implication.

Chapter 5 applies some of the techniques of the previous chapters to the computation of exactly provable formulas in provability logic. This chapter is a revised version of [HJ 96], with an emphasis on the contributions of the present author, viz. the introduction of semantic types and the computation of the exactly provable formulas with no nesting of the provability operator.

Chapter 6 describes a family of theorem testers based on semantic tableaux, including a tester for **IpL** and testers for several modal logics and contains a description of the algorithms to compute diagrams and exact models.

Appendix A contains some of the more important parts of the computer programs (in the programming language C), that are described in this thesis.

Appendix B is a collection of examples of the output of the computer programs that calculated diagrams of fragments and the exactly provable formulas in $\mathbf{L}_1^1$.

Appendix C contains tables of the number of equivalence classes computed for several fragments of **IpL**.

1.2 General preliminaries

The language of classical propositional logic (**CpL**), intuitionistic propositional logic (**IpL**) and modal propositional logic used in the consecutive chapters consists of the constants $\bot$ and $\top$ and an infinite stock of propositional variables $\{p_1, p_2, \ldots\}$, also called atomic formulas (or simply atoms), together with the usual propositional connectives $\{\wedge, \vee, \rightarrow, \neg\}$ (and sometimes $\neg\neg$). In the case of modal logic also $\Box$ and $\Diamond$ are included. We will, in the case of classical (modal) logic, most of the time treat $\vee, \rightarrow$ and $\Diamond$ as defined from $\wedge, \neg$ and $\Box$ (but sometimes $\vee, \rightarrow$ and $\Box$ defined from $\wedge, \neg$ and $\Diamond$). On the other hand, in **IpL** we will take $\neg\phi$ to be defined as $\phi\rightarrow\bot$.

To avoid a plethora of parentheses, in writing our formulas, we define the order in which the connectives take preference above each other as:

$$\Diamond \qquad \Box \qquad \neg \qquad \wedge \qquad \vee \qquad \rightarrow.$$

Hence, $\neg\Box\neg p\rightarrow q \wedge r \vee s$ is equivalent to $\neg(\Box(\neg p))\rightarrow((q \wedge r) \vee s)$.

The derivability relation for a logic L will be denoted by $\vdash_L$ or by $\vdash$ if the choice of the logic is obvious from the context. Formulas ϕ and ψ are called *equivalent* in the logic L, $\phi \equiv_L \psi$ (or $\phi \equiv \psi$ if L is obvious), if they are interderivable: $\phi \vdash_L \psi$ and $\psi \vdash_L \phi$. The derivability relations of the logics treated in the sequel are assumed to be defined by the set of rules and axioms below. Let T and T' be sets of formulas and ϕ, ψ and χ be formulas. We will define $\vdash$ as a relation between sets of formulas and formulas, but we will write $T, \phi \vdash \psi$, where more formally $T \cup \{\phi\} \vdash \psi$ is meant.

The rules for intuitionistic propositional logic are:

1. $\phi \in T \quad \Rightarrow \quad T \vdash \phi$
2. $T, \psi \vdash \phi$ and $T' \vdash \psi \quad \Rightarrow \quad T \cup T' \vdash \phi$
3. $T \vdash \phi$ and $T \vdash \psi \quad \Leftrightarrow \quad T \vdash \phi \wedge \psi$
4. $T, \phi \vdash \chi$ and $T, \psi \vdash \chi \quad \Leftrightarrow \quad T, \phi \vee \psi \vdash \chi$
5. $T, \phi \vdash \psi \quad \Leftrightarrow \quad T \vdash \phi\rightarrow\psi$
6. $\bot \in T \quad \Rightarrow \quad T \vdash \phi$

In the case of classical logic we add the axiom:

7. $\neg\neg\phi \vdash \phi$

In the case of the classical modal logic **K** we add also:

8. $\vdash \phi \quad \Rightarrow \quad \vdash \Box\phi$
9. $\Box(\phi \rightarrow \psi) \vdash \Box\phi \rightarrow \Box\psi$

For classical and intuitionistic propositional logic, alternative axioms and rules can also be found for example in [TD 82] and for (other) modal logics one may consult [HC 84].

A *fragment* is a sublanguage of a logic, obtained by restricting the set of atoms or the application of connectives (or both). In this thesis we will often restrict the language to a finite set of atoms $p_1, \ldots, p_n$. Let F and G be fragments of a logic L. Then G is called a *subfragment* of F, if every formula of G is a formula of F (which will be denoted as $G \subseteq F$). The *diagram*, $Diag(F)$, of a fragment F, is the set of equivalence classes in F ordered by $\vdash$.

Let $\langle W, \leq \rangle$ be an ordered set (more traditionally: a partially ordered set or poset). A subset $X \subseteq W$ will be called a *closed* subset of W if for all v and w in W, $v \in X$ and $v \leq w$ implies $w \in X$. The set of closed subsets of W will be denoted by $\mathcal{P}^*(W)$.

The Kripke model theory used here is fairly standard and can be found, for example, in [Benthem 83]. A *Kripke frame* is a tuple $\langle W, R \rangle$, with a *domain* W (the set of *worlds* or *nodes*) and $R \subseteq W^2$ a binary relation on W. The relation R is called an *accessibility relation*. If the accessibility relation is known to be irreflexive, we will often use $<$ for R. If R is reflexive, we will use $\leq$ and $l < k$ (or $k > l$) will be used as a shorthand for $l \leq k$ and $l \neq k$. lRk will sometimes also be written as $k\breve{R}l$. If k and l are nodes in W and kRl, then l is called a *successor* of k and k is called a *predecessor* of l. A node l is a *direct successor* of k, kR_1l (or $k <_1 l$), if $k \neq l$ and for all m such that kRm and mRl either $m = k$ or $m = l$. A node k is the *root* of a Kripke model K if k is the only node in K that has at most itself as a predecessor. A node k is a *terminal* node if k has at most itself as a successor. So a terminal node has only itself as a successor or no successors at all.

A *Kripke model* $K = \langle W, R, atom \rangle$ is a Kripke frame $\langle W, R \rangle$ with a valuation *atom*, mapping nodes of W to sets of propositional variables. As usual we will define $k \Vdash \phi$, the *forcing* of a formula ϕ by a node k in a Kripke model K.

The Kripke models defined above will be used in (classical) modal logic. For **CpL** and **IpL** we will define special classes of Kripke models.

1.2.0.1. DEFINITION. *Let $K = \langle W, R, atom \rangle$ be a Kripke model.*
K is a **CpL** Kripke model *if R is the identity relation.*
K is an intuitionistic Kripke model **IpL** *Kripke model for short) if*

1. *R is reflexive, transitive and anti-symmetric;*
2. *atom is order preserving.*

We will use $atom^n$ for the restriction of *atom* to $\{p_1, \ldots, p_n\}$ (nowhere a $q \notin \{p_1, \ldots, p_n\}$ is forced). Note that $K = \langle W, R, atom^n \rangle$ is again a Kripke model, which will be called an *n-model*.

In the sequel we will write $k \in K$ to express that a world k is a node of Kripke model K, instead of the more pedantic $K = \langle W, R, atom\rangle$ and $k \in W$. If $K = \langle W, R, atom\rangle$ and $V \subseteq W$ then $L = \langle V, R{\restriction}V, atom{\restriction}V\rangle$ is called a *submodel*. The fact that L is a submodel of K will be written as $L \subseteq K$.

Let us first recall the truth definition of classical propositional logic, **CpL**, in terms of Kripke semantics.

1.2.0.2. DEFINITION. *Let $K = \langle W, R, atom\rangle$ be a Kripke model and $k \in K$. Define $k \Vdash \phi$, the node k forces the formula ϕ, inductively as:*

- $k \Vdash p \quad \Leftrightarrow \quad p \in atom(k)$ *(p atomic);*
- $k \Vdash \psi \wedge \chi \quad \Leftrightarrow \quad k \Vdash \psi$ *and* $k \Vdash \chi$;
- $k \Vdash \psi \vee \chi \quad \Leftrightarrow \quad k \Vdash \psi$ *or* $k \Vdash \chi$;
- $k \Vdash \psi \rightarrow \chi \quad \Leftrightarrow \quad k \nVdash \psi$ *or* $k \Vdash \chi$;
- $k \Vdash \neg\psi \quad \Leftrightarrow \quad k \nVdash \psi$ *(i.e not $k \Vdash \psi$).*

We will say that K models *ϕ ($K \Vdash \phi$) (or ϕ holds in K) if for all $k \in K$ it is true that $k \Vdash \phi$.*

To obtain the Kripke semantics for modal logic we need to add rules for the modal operators to the rules defined for **CpL**.

1.2.0.3. DEFINITION. *Let $K = \langle W, R, atom\rangle$ be a Kripke model and $k \in K$.*

- $k \Vdash \Box\psi \quad \Leftrightarrow \quad \forall l \in K($*if* kRl *then* $l \Vdash \psi)$;
- $k \Vdash \Diamond\psi \quad \Leftrightarrow \quad \exists l \in K(kRl$ *and* $l \Vdash \psi)$;

Again K *models* ϕ ($K \Vdash \phi$) (or ϕ holds in K) if for all $k \in K$ it is true that $k \Vdash \phi$.

Next we define the forcing relation on intuitionistic Kripke models. Note that in the Kripke semantics of **IpL** implication and negation have non-local behaviour, like the modal operators.

1.2.0.4. DEFINITION. *Let $K = \langle W, R, atom\rangle$ be an intuitionistic Kripke model and let $k \in K$. Define $k \Vdash \phi$, inductively as:*

- $k \Vdash p \quad \Leftrightarrow \quad p \in atom(k)$ *(p atomic);*
- $k \Vdash \psi \wedge \chi \quad \Leftrightarrow \quad k \Vdash \psi$ *and* $k \Vdash \chi$;
- $k \Vdash \psi \vee \chi \quad \Leftrightarrow \quad k \Vdash \psi$ *or* $k \Vdash \chi$;
- $k \Vdash \psi \rightarrow \chi \quad \Leftrightarrow \quad \forall l \in K($*if* kRl *then* $l \nVdash \psi$ *or* $l \Vdash \chi)$;
- $k \Vdash \neg\psi \quad \Leftrightarrow \quad \forall l \in K($*if* kRl *then* $l \nVdash \psi)$.

And as above, K *models* ϕ ($K \Vdash \phi$) (or ϕ holds in K) if for all $k \in K$ it is true that $k \Vdash \phi$.

Let $\mathcal{M}$ be a class of Kripke models. A formula ψ is a *local $\mathcal{M}$-consequence* of a formula ϕ, $\phi \models_{\mathcal{M}} \psi$, if for every $K \in \mathcal{M}$ and every $k \in K$ such that $k \Vdash \phi$ it is true that $k \Vdash \psi$. A formula ψ is a *global $\mathcal{M}$-consequence* of a formula ϕ, $\phi \Vdash_{\mathcal{M}} \psi$, if for every $K \in \mathcal{M}$ such that $K \Vdash \phi$ it is true that $K \Vdash \psi$.

A propositional logic L is said to be *sound* for $\mathcal{M}$ if for all L-formulas ϕ and ψ: $\phi \vdash_L \psi$ implies $\phi \models_{\mathcal{M}} \psi$. L is said to be *complete* for $\mathcal{M}$, if for all L-formulas ϕ and

ψ such that $\phi \models_{\mathcal{M}} \psi$ it is true that $\phi \vdash_L \psi$. If we restrict our attention to formulas in a logic L with all atomic subformulas in the set $\{p_1, \ldots, p_n\}$, we obtain the fragment L^n. If L is sound and complete for a class of Kripke models $\mathcal{M}$, one can, usually with almost the same proof, obtain also a theorem stating that L^n is sound and complete for the n-models in $\mathcal{M}$ (i.e. L is n-complete for $\mathcal{M}$). The following well known soundness and completeness theorems are stated here as facts[3].

1.2.0.5. FACTS.

1. **K** *is sound and complete for the class of finite Kripke models.*
2. **CpL** *is sound and complete for the class of finite* **CpL** *Kripke models.*
3. **IpL** *is sound and complete for the class of finite* **IpL** *Kripke models.*

For the proofs of these facts we refer to [HC 84] and [TD 88].

If $L \subseteq K$ then L is a *generated submodel* of K if the domain of L is a closed subset of K. As a notation for the generated submodel above a node we will use $\uparrow k = \{l \mid kRl\}$. For the smallest generated submodel including node k, we will use the notation $\underline{\uparrow} k = \{l \mid kRl \text{ or } l = k\}$. If the accessibility relation is known to be reflexive, then of course $\uparrow k = \underline{\uparrow} k$ for all nodes $k \in K$. Occasionally we will use $\downarrow k$ for the set of nodes below node k, hence $\downarrow k = \{l \mid lRk\}$.

1.2.0.6. DEFINITION. *Let K be a Kripke model. The nodes $k_1, \ldots, k_n \in K$ form a* cycle *in K of length n if $k_n R k_1$ and $1 \leq i < n$ implies $k_i R k_{i+1}$.*

K is called anticyclic *if K contains no cycles of length more than 1.*

In a finite anticyclic Kripke model K, we define the *depth* of a node $k \in K$ as usual.

1.2.0.7. DEFINITION. *If K is a finite anticyclic Kripke model and k is a node of K, then $\delta(k)$, the* depth *of k is defined as*

$$\delta(k) = \begin{cases} 0 & \textit{if } kRl \textit{ implies } k = l \\ \max\{\delta(l) \mid l \neq k \textit{ and } kRl\} + 1 & \textit{otherwise.} \end{cases}$$

Most of the models in this thesis will be Kripke models. The next definition however introduces a more abstract notion of model. This allows us to call a finite representation of a fragment a *model* even if it is not a Kripke model. As we will see in Chapter 3, not all exact models are exact Kripke models.

[3] If we designate a proposition as a fact, we will not give a proof, either because it can be found elsewhere, or because it is rather trivial.

1.2.0.8. DEFINITION. *A structure* $M = \langle W, \preceq, \omega \rangle$ *is called a* model *for fragment* F *if* $\langle W, \preceq \rangle$ *is an ordered set and* $\omega : F \mapsto \mathcal{P}^*(W)$*, such that for all formulas* $\phi, \psi \in F$*:*

$$\phi \vdash \psi \quad \Rightarrow \quad \omega(\phi) \subseteq \omega(\psi).$$

M is called a classical *model if $\preceq$ is the identity relation (and hence all subsets of W are closed: $\mathcal{P}^*(W) = \mathcal{P}(W)$).*

A model M is complete *for F if for all ϕ and ψ in F*

$$\omega(\phi) \subseteq \omega(\psi) \quad \Rightarrow \quad \phi \vdash \psi.$$

A model M is exact *for F if M is complete for F and ω is surjective.*

Note that the definition the valuation of these abstract models does not require recursion on the length of the formula ϕ. The valuation ω may be any kind of mapping from formulas into closed subsets of W, as long as ω is monotone in the order of derivability.

A Kripke model corresponds to a model $\langle W, \preceq, \omega \rangle$ in the sense of the definition above. In classical models the relation $\preceq$ will be the identity, and in intuitionistic models $\preceq$ coincides with R. In both cases the function $[\![\phi]\!] = \{k \in K \mid k \Vdash \phi\}$ will map formulas onto (closed) subsets of K, in such a way, that $\phi \vdash \psi \quad \Rightarrow \quad [\![\phi]\!] \subseteq [\![\psi]\!]$.

Suppose $M = \langle W, \preceq, \omega \rangle$ is a model for fragment F. For $w \in W$ and $\phi \in F$ we define $w \Vdash \phi$, in a natural way, by

$$w \Vdash \phi \quad \Leftrightarrow \quad w \in \omega(\phi).$$

Obviously if $\phi \vdash \psi$ and $w \Vdash \phi$ we may infer that $w \Vdash \psi$.

This definition includes models with $W \subseteq F$, $\preceq$ the restriction of $\dashv$, the converse of $\vdash$, to W and ω defined as $\omega(\phi) = \{\psi \in W \mid \psi \vdash \phi\}$.

Note that if $M = \langle W, \preceq, \omega \rangle$ is an exact model of fragment F, then

$$\langle \mathcal{P}^*(W), \subseteq \rangle \cong Diag(F).$$

As it is our intention to use Kripke semantics as a general framework for the semantics of **CpL**, **IpL** as well as modal logics, it may be worthwhile to define forcing of a formula by a node in a Kripke model with respect to a general language containing the languages of these logics.

For example, in our computer programs for calculating diagrams of fragments from exact models there is only a small difference between modal logic and intuitionistic logic, as will be pointed out in Chapter 2. The rules for calculating the set of worlds in the exact model that force a certain formula ϕ can easily be explained using the generalized language and its Kripke semantics.

This generalized language can be defined as a language of propositional modal logic, with the connectives $\Diamond, \sim, \wedge, \vee, \Rightarrow$ and constants $\bot$ and $\top$.

The definition of $k \Vdash \phi$, a node k forcing a formula ϕ, is as definition 1.2.0.2 and 1.2.0.3 for $\Diamond, \wedge$ and $\vee$. The definition below of forcing for $\sim$ and $\Rightarrow$ reveals that $\sim$ is the classical negation and $\Rightarrow$ the intuitionistic implication:

1.2.0.9. DEFINITION.

1. $k \Vdash \psi \Rightarrow \chi \quad \Leftrightarrow \quad \forall l \in K(\textit{if}\, kRl \textit{ then } l \nVdash \psi \textit{ or } l \Vdash \chi)$;
2. $k \Vdash \sim \psi \quad \Leftrightarrow \quad k \nVdash \psi$;

where $k \nVdash \psi$ *is shorthand for* not $k \Vdash \psi$.

We can turn our generalized language into the language of classical modal propositional logic, by removing $\Rightarrow$, defining $\neg$ as $\sim$ and defining $\phi \to \psi$ and $\Box$ as usual as $\neg\phi \vee \psi$ and $\neg\Diamond\neg$. For the language of **CpL** we have to remove $\Diamond$ and $\Box$ from the language of classical modal logic. Likewise we can define the language of **IpL** by removing $\Diamond$ and $\sim$ from the generalized language, defining $\to$ as $\Rightarrow$ and defining $\neg\phi = \phi \to \bot$.

Chapter 2

Semantic Types and Exact Models

2.1 Introduction

In this chapter we will introduce the notion of the *semantic type* of a world in a Kripke model and explain the relation between semantic types, type formulas and exact models. Within this framework we will restate some proofs of related classical theorems about **CpL**, **K** and **IpL**. In the next chapters we will use semantic types to obtain exact models of finite fragments of **IpL** and calculate axioms of interpreted theories of provability logic.

A semantic type, in some fragment F of modal or intuitionistic propositional logic, is an abstract representation of a node in a Kripke model. The idea is that the semantic type of a node k in a Kripke model contains exactly the information that determine which formulas are forced in k, i.e. if a node l has the same semantic type as k in F, then k and l force the same formulas in F. We will write $Th_F(k)$ for the F-theory of k, defined by $Th_F(k) = \{\phi \in F \mid k \Vdash \phi\}$. In analogy to the situation in model theory, cf. [CK 73], $Th_F(k)$ could be called the *type* of k (in the language of F). If F is finite and closed under $\wedge$, there is formula $\phi_F(k)$, unique up to equivalence, which is a conjunction of representatives of every equivalence class in $Th_F(k)$. Such a formula $\phi_F(k)$ is an axiom for $Th_F(k)$ and is written as $\phi_F(k) \equiv \bigwedge Th_F(k)$. Here we will adopt the terminology of modal logic and call the formula $\phi_F(k)$ (or more precisely its equivalence class) the F-*type* of k (or the *type formula* of k in F). In modal logic (especially in provability logic) types are also known as the *character* of k (in [Bernardi 75] and [GG 90] or as an *atom* in [Bellissima 84]). The term *type* was used in [Shavrukov 93].

For the relation between types and exact Kripke models, recall definition 1.2.0.8. A Kripke model K is an exact Kripke model for a fragment F if K has the following properties:

1. K is F-complete, i.e. for all $\phi, \psi \in F$:

$$\phi \vdash \psi \quad \Leftrightarrow \quad [\![\phi]\!] \subseteq [\![\psi]\!].$$

2. Every closed subset of K is F-definable, i.e. for all closed $X \subseteq K$ there is a $\phi \in F$ such that:
$$X = [\![\phi]\!].$$

Observe that by property 2, for every k in an exact Kripke model there is a type formula for k in F.

For a finite representation of the fragment F, the types in the exact Kripke model of F would be sufficient. Recall the generalized notion of model from definition 1.2.0.8. The set of types T (ordered by derivability) is an *exact model* if we add the mapping ω, defined by

$$\omega(\phi) = \{\psi \in T \mid \psi \vdash \phi\}.$$

If a fragment F has an exact model, the type formulas in an exact model for F can often be derived from some *normal form* for the formulas in F. But such an exact model with a set of formulas as its universe, is less useful for our purpose than an exact Kripke model. In the calculation of $\omega(\phi)$ in a general exact model, one uses the derivability relation, instead of deciding the derivability of ψ from ϕ by model checking, as in an exact Kripke model.

On the other hand, the general exact model almost gives us an exact Kripke model. We only have to construct a Kripke model K such that for each type ϕ in the general exact model there is a $k \in K$ such that ϕ is the type of k (and such that this mapping of types on worlds of K is 1–1). The core question for this step in the construction of an exact Kripke model is: 'which kind of world does realize type ϕ'.

The answer to this question is a semantic equivalent to the type formula of a node k, and will be called the *semantic type* of k in F. The semantic type is an abstract representation of the node, in such a way that identical semantic types in F are equivalent in F (i.e. have the same F-theory).

To represent the essential information about a node k in a Kripke model we have to know:

a. which atoms hold in k;
b. what happens in the successor nodes.

If we use $\tau_F(k)$ for the semantic type of k in K, the general format of a semantic type is
$$\tau_F(k) = \langle atom^n(k), T\rangle$$
where T is a set of semantic types of successors[1] of k in K.

Of course, the semantic types of F have to fulfill the condition:

$$\tau_F(k) = \tau_F(l) \quad \Leftrightarrow \quad Th_F(k) = Th_F(l)$$

[1] As we will see in Chapter 3, sometimes the information about a subset of the successors of k is sufficient.

for all $k \in K$ and $l \in L$ where K and L are Kripke models used in the semantics of F. If there is a type formula for each node k in F, a $\phi_F(k) \in F$ that is an axiom[2] for $Th_F(k)$, the condition above is equivalent to:

$$\tau_F(k) = \tau_F(l) \quad \Leftrightarrow \quad \phi_F(k) = \phi_F(l).$$

Of course the distinction between the semantic types in F and the $\phi_F(k)$ is just a matter of point of view.

Note that with the restrictions that apply to the fragment F, regarding the atoms used and the applicability of connectives, the information we need in the semantic type of k in K need not be a full description of the submodel of K generated by k. Otherwise we would not have gained much in switching from Kripke models to semantic types.

The approach in this thesis to the construction of the exact model of a fragment F will be to find a minimal set of semantic types that is *complete* for F. First we will define what kind of objects the semantic types for F are (in some class of Kripke models). Next we will define a set T of these semantic types such that for each ϕ and ψ in F with $\phi \nvdash \psi$ there is a type $t \in T$ available, such that if for a node k in a Kripke model K, $\tau_F(k) = t$, then $k \Vdash \phi$ and $k \nVdash \psi$.

If we prove T to be minimal, the construction of our exact Kripke model is almost complete because the semantic types usually carry in them a natural accessibility relation. However in modal logic this order relation is not unique, the semantic types may be ordered in various alternative ways to obtain an exact Kripke model.

Turning to intuitionistic propositional logic, a fragment F will have a finite exact model iff $Diag(F)$ is isomorphic to a set of closed subsets ordered by inclusion, as was pointed out in the preliminary section of the introduction. Hence, $Diag(F)$ is a finite distributive lattice. Let us use $\phi \oplus \psi$ for the representative of the equivalence class in $Diag(F)$ that is the join of the classes represented by ϕ and ψ. Note that if $\vee$ is one of the connectives of F then $\phi \oplus \psi \equiv \phi \vee \psi$.

2.1.0.1. DEFINITION. *An equivalence class ϕ will be called* join-irreducible *(or*irreducible *for short) in $Diag(F)$ if ϕ is not the bottom element of $Diag(F)$ and for all $\psi, \chi \in F$ we have*

$$\phi \vdash \psi \oplus \chi \quad \Rightarrow \quad \phi \vdash \psi \text{ or } \phi \vdash \chi.$$

Let us denote the set of join-irreducible classes in $Diag(F)$ as $I(F)$. Then $I(F)$ may be regarded as an ordered set (with $\dashv$, the reverse of $\vdash$, as its order relation[3]) and hence the set of closed subsets $\mathcal{P}^*(I(F))$ is defined. According to Birkhoff's representation theorem $Diag(F)$ will be isomorphic to $\mathcal{P}^*(I(F))$ ordered by inclusion.

[2]As is the case if F is finite.

[3]This is more convenient in case we want to turn $I(F)$ into a Kripke model.

2.1.0.2. THEOREM. (Birkhoff) *Any finite distributive lattice is isomorphic to the lattice of the closed sets of its join-irreducible elements.*

Proof. A proof can be found in [DP 90]. ⊣

Hence $I(F)$ will be a set of *types* that can be used for an exact model. However in the intuitionistic case the order of the exact model, given by ⊣, will in general not be the identity relation. (In classical logic, where we deal with Boolean algebras instead of Heyting algebras, the different atoms of the algebra exclude each other: if ϕ and ψ are irreducible, then $\phi \vdash \psi \quad \Rightarrow \quad \phi \equiv \psi$.)

In intuitionistic propositional logic the general notion of exact model is closer to that of an exact Kripke model than in classical modal logic. It is not difficult to turn an exact model for F into a Kripke model by stipulating the valuation $atom(\phi) = \{p \text{ atomic} \mid \phi \vdash p\}$.

However, these notions do not coincide, as we cannot prove in general for the resulting Kripke model that the node corresponding to type ϕ does indeed force the formula ϕ. As we will see in Chapter 3 the fragment $[\vee, \neg]^n$ has for each n an exact model, which is, for $n > 1$, not an exact Kripke model.

As the order in an exact Kripke model of **IpL** is induced by the derivability relation, the exact Kripke model of a fragment F, if it exists, is unique (that is, isomorphic to the set of irreducibles $I(F)$ ordered by derivability).

If, as in the case of the $[\vee, \neg]^n$ fragments, an exact Kripke model is out of reach, but we do have a minimal complete set of semantic types for the fragment F at hand, we can at least try to find a minimal finite model realizing all of the types in this set. The resulting model is of course complete for the fragment F and will be called a *universal model* for F.

2.2 Preliminaries

In this section we will introduce some useful notations for semantic types, introduce the important notion of *bisimulation* between Kripke models and define a layered variant of this relation. These layered bisimulations are related to the *model equivalence* in [Fine 74] and play a major rôle in the semantics of fragments with restricted nesting of modal operators or restricted nesting of implication.

We will take the liberty of using R for the accessibility relation even in those cases where we are dealing with more than one model. Usually it is clear from the context which model the relation belongs to.

2.2.0.1. DEFINITION. *A relation S between two Kripke models K and L is said to be a* bisimulation *iff for all $k \in K$ and $l \in L$ such that kSl:*

1. $atom(k) = atom(l)$;
2. $\forall k' \breve{R} k \; \exists l' \breve{R} l \; (k'Sl')$;
3. $\forall l' \breve{R} l \; \exists k' \breve{R} k \; (k'Sl')$.

A bisimulation relation which is a function is called a p-morphism. *If a p-morphism from K to L is surjective, it is often called a* reduction *from K to L (also known as pseudo-epimorphism).*

We will use the notation $k \rightleftarrows l$ to denote that k and l bisimulate each other, that is, there exists a non-empty bisimulation S such that kSl. That $\rightleftarrows$ is an equivalence relation between nodes in Kripke models is obvious.

2.2.0.2. THEOREM. (Bisimulation Theorem) *If* $k \rightleftarrows l$ *and* $k \Vdash \phi$ *then* $l \Vdash \phi$.

Proof. For propositional formulas ϕ, both in modal logic and intuitionistic logic, the theorem is easily proved by induction on the length of ϕ. Note that we could use the general language from the general preliminaries to prove this theorem for both logics at once. ⊣

An *n-bisimulation* is a bisimulation between two n-models (and hence with the first condition of definition 2.2.0.1 changed into: $atom^n(k) = atom^n(l)$). If k and l n-bisimulate each other we will write $k \rightleftarrows^n l$.

Spelling out the proof of the bisimulation theorem will reveal that it can easily be transformed into a proof that if all propositional variables of ϕ are in $\{p_1, \ldots, p_n\}$, then $k \rightleftarrows^n l \quad \Rightarrow \quad k \Vdash \phi$ then $l \Vdash \phi$.

Layered bisimulations also known as *bounded bisimulations* or n, m-bisimulations, will be defined by induction on m.

2.2.0.3. DEFINITION. *A relation S between two Kripke models K and L is said to be an* $n, 0$-bisimulation *iff for all $k \in K$ and $l \in L$ such that kSl it is true that $atom^n(k) = atom^n(l)$.*

*A relation S between two Kripke models K and L is said to be an $n, m+1$-*bisimulation *iff there is a n, m-bisimulation S' such that, for all $k \in K$ and $l \in L$ with kSl,*

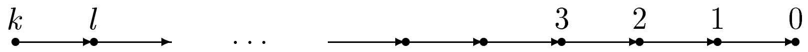
1. $atom^n(k) = atom^n(l)$;
2. $\forall k' \breve{R} k \; \exists l' \breve{R} l \; (k' S' l')$;
3. $\forall l' \breve{R} \, l \exists k' \breve{R} k \; (k' S' l)'$.

We will write $k \rightleftarrows^n_m l$ if there exists an n, m-bisimulation between k and l.

In the section on modal logic in this chapter, we will prove an n, m-bisimulation theorem for fragments of modal logic with atomic formulas in $\{p_1, \ldots, p_n\}$ and modal degree less than or equal to m. In Chapter 4 a similar theorem is proved for $\mathbf{IpL}^n_m$, the fragment with atomic formulas in $\{p_1, \ldots, p_n\}$ and the nesting of implication bounded by m. Fragments with this kind of restriction on the nesting of one of the connectives are called *layered fragments.*

It will be clear from these definitions that $k \rightleftarrows l$ implies $k \rightleftarrows^n l$, which implies $k \rightleftarrows^n_m l$ for each m. Our notation may suggest that $k \rightleftarrows^n l$ if $\forall m \; (k \rightleftarrows^n_m l)$. In general this is not true, as the following counter-example shows.

2.2.0.4. Example. *In the Kripke model below the accessibility relation is irreflexive. At the right hand side of l there is a copy of the natural numbers in descending order. No atoms are forced in the nodes of this model. We have $k \mathrel{\underline{\leftrightarrow}}^n_m l$ for each n and m, but not $k \mathrel{\underline{\leftrightarrow}}^n l$.*



2. Figure. *A counter-example against $\forall m\ (k \mathrel{\underline{\leftrightarrow}}^n_m l) \quad \Rightarrow \quad k \mathrel{\underline{\leftrightarrow}}^n l$.*

In case k and l are nodes in finite Kripke models, $\forall m\ (k \mathrel{\underline{\leftrightarrow}}^n_m l)$ does imply $k \mathrel{\underline{\leftrightarrow}}^n l$.

2.2.0.5. Definition. *A Kripke model K is called* locally finite *if for every node $k \in K$ the set $\uparrow k = \{l \in K \mid kRl\}$ is finite.*

2.2.0.6. Theorem. *For nodes k and l in locally finite Kripke models:*

$$\forall m\ (k \mathrel{\underline{\leftrightarrow}}^n_m l) \quad \Leftrightarrow \quad k \mathrel{\underline{\leftrightarrow}}^n l.$$

Proof. Assume $k \mathrel{\underline{\leftrightarrow}}^n_m l$ for all m. We will prove that the relation S between the (finite) models of k and l defined as $k'Sl' \quad \Leftrightarrow \quad \forall m (k' \mathrel{\underline{\leftrightarrow}}^n_m l')$ is a bisimulation.

That $atom^n(k) = atom^n(l)$, is an immediate consequence of the definition of n,m-bisimulation. As the other two conditions for a bisimulation are symmetric, we only prove the first.

Suppose kRk' and let $l_1, \ldots, l_r$ be an enumeration of the successors of l. We will prove that there is an $i \leq r$ such that $k' \mathrel{\underline{\leftrightarrow}}^n_m l_i$ for all m.

For every $i \leq r$ such that not $\forall m\ (k' \mathrel{\underline{\leftrightarrow}}^n_m l_i)$ there is a least m, say m_i, with not $k' \mathrel{\underline{\leftrightarrow}}^n_{m_i} l_i$.

If not $\exists l' \breve{R} l\ \forall m\ (k' \mathrel{\underline{\leftrightarrow}}^n_m l')$, then let $M = \max\{m_i \mid m_i = \min\{m \mid \text{not } k' \mathrel{\underline{\leftrightarrow}}^n_m l_i\}\}$. Hence for no l_i it will be true that $k' \mathrel{\underline{\leftrightarrow}}^n_M l_i$, for it is easy to prove from the definition of n,m-bisimulation, that $k' \mathrel{\underline{\leftrightarrow}}^n_M l_i$ would imply $k' \mathrel{\underline{\leftrightarrow}}^n_m l_i$ for all $m \leq M$.

By assumption we know $k \mathrel{\underline{\leftrightarrow}}^n_{M+1} l$ and hence for some $l' \breve{R} l$ it should be true that $k' \mathrel{\underline{\leftrightarrow}}^n_M l'$. From this contradiction we infer that for some $l_i \breve{R} l$ it is true that $\forall m\ (k' \mathrel{\underline{\leftrightarrow}}^n_m l_i)$. ⊣

2.2.0.7. Corollary. *For nodes k and l in finite Kripke models:*

$$\forall m\ (k \mathrel{\underline{\leftrightarrow}}^n_m l) \quad \Leftrightarrow \quad k \mathrel{\underline{\leftrightarrow}}^n l.$$

As stated in the introduction of this chapter, the semantic type of a node k in a Kripke model K will in general be of the form $\tau(k) = \langle atom(k), T \rangle$, where T a set of types of successors of k. To point out the separate parts of a type we also define the projections $j_0(t)$ and $j_1(t)$ for a tuple t.

2.2.0.8. DEFINITION. *Let t be a tuple, $t = \langle P, Q\rangle$. Define $j_0(t) = P$ and $j_1(t) = Q$.*

In layered fragments (with restricted nesting of modal operators or implication) we will introduce hierarchies of types, $\{T_m \mid m \in I\!N\}$. If $t \in T_0$ then $j_1(t) = \emptyset$ holds, while for $t \in T_{m+1}$ we will have $j_1(t) \subseteq T_m$.

If $\tau_F(k) = t$, then k is said to *realize* the type t.

2.3 Types in CpL

The main point of introducing (semantic) types and exact models in classical propositional logic, **CpL**, is to illustrate the concepts defined in the introduction of this chapter. Some facts about **CpL** and its types that appear in this section are also useful in the next sections and chapters.

Recall that $\mathbf{CpL}^n$ is the fragment of **CpL** formulas of which the atomic subformulas belong to the set $\{p_1, \ldots, p_n\}$. By the n-completeness theorem a $\mathbf{CpL}^n$ formula ϕ is derivable in **CpL** iff ϕ is valid in all finite n-models K.

2.3.0.1. DEFINITION. *Let $Q \subseteq \{p_1, \ldots, p_n\}$ be a finite set of atoms. Define:*

$$\phi^n_Q = \bigwedge Q \wedge \bigwedge\{\neg q \mid q \in \{p_1, \ldots, p_n\} \setminus Q\}.$$

The definition of the formulas ϕ^n_Q will also be useful in later chapters.

2.3.0.2. DEFINITION. *The* type $\phi^n_{\mathbf{CpL}}(k)$ *of a world k in an n-model K is the formula $\phi^n_{atom^n(k)}$.*

Only in this section we will write $\phi^n(k)$ for $\phi^n_{\mathbf{CpL}}(k)$. In other fragments where the **CpL** type of a node is used it will be necessary to distinguish the type $\phi^n_{\mathbf{CpL}}(k)$ from the type $\phi^n(k)$ in the fragment at hand.

If k is a world in a **CpL** model, let us write $Th^n(k)$ for the $\mathbf{CpL}^n$ theory of k, defined by $Th^n(k) = \{\phi \in \mathbf{CpL}^n \mid k \Vdash \phi\}$.

2.3.0.3. LEMMA. *Let k be a node in an n-model and let $\phi^n(k)$ be the type of k in $\mathbf{CpL}^n$. Then*

1. *ϕ is an irreducible formula in $\mathbf{CpL}^n$ (see definition 2.1.0.1) iff for all formulas $\psi \in \mathbf{CpL}^n$:*
$$\phi \nvdash \psi \quad \Leftrightarrow \quad \phi \vdash \neg\psi.$$
2. *$\phi^n(k)$ is irreducible, i.e.:*
$$\forall \psi, \chi \in \mathbf{CpL}^n(\phi^n(k) \vdash \psi \vee \chi \quad \Rightarrow \quad \phi^n(k) \vdash \psi \text{ or } \phi^n(k) \vdash \chi).$$
3. *if a $\mathbf{CpL}^n$ formula ϕ is irreducible (in $\mathbf{CpL}^n$), then ϕ is equivalent to a type of $\mathbf{CpL}^n$;*
4. *the Lindenbaum algebra of $\mathbf{CpL}^n$ is a finite Boolean algebra with the types of $\mathbf{CpL}^n$ as its atoms;*

5. *if l is a node in an n-model K, then*

$$l \Vdash \phi^n(k) \quad \Leftrightarrow \quad atom(l) = atom(k).$$

6. *$\phi^n(k)$ is an axiom for $Th^n(k)$.*

Proof. 1: Let $\phi \in \mathbf{CpL}^n$ be irreducible. Then from $\phi \vdash \psi \vee \neg\psi$ infer that $\phi \vdash \psi$ or $\phi \vdash \neg\psi$. For the other direction, let $\phi \vdash \psi \vee \chi$. If $\phi \nvdash \psi$, then by assumption, $\phi \vdash \neg\psi$. Hence we would have $\phi \vdash \chi$.

2: With a simple induction on the length of formula $\psi \in \mathbf{CpL}^n$ prove that $\phi^n(k) \vdash \psi$ or $\phi^n(k) \vdash \neg\psi$.

3: Let $\phi \in \mathbf{CpL}^n$ be irreducible. According to definition 2.1.0.1 $\phi \nVdash \bot$. Hence, for some node k in a $\mathbf{CpL}$ n-model, we have $k \Vdash \phi$. Now note that obviously $k \Vdash \phi^n(k)$, hence $\phi^n(k) \nvdash \neg\phi$ and $\phi \nvdash \neg\phi^n(k)$. As both ϕ and $\phi^n(k)$ are irreducible, this implies $\phi \equiv \phi^n(k)$.

4: To prove $\phi^n(k)$ to be an atom in $Diag(\mathbf{CpL}^n)$, assume that $\phi \in \mathbf{CpL}^n$, $\phi \not\equiv \bot$ and $\phi \vdash \phi^n(k)$. As $\phi^n(k)$ is irreducible, use 1 to infer from $\phi^n(k) \nvdash \neg\phi$ that $\phi^n(k) \equiv \phi$.

5: By definition, if $atom^n(k) = atom^n(l)$, then $\phi^n(k) = \phi^n(l)$. For the other direction, assume that $l \Vdash \phi^n(k)$. Then both $\phi^n(k) \nvdash \neg\phi^n(l)$ and $\phi^n(l) \nvdash \neg\phi^n(k)$. From the irreducibility of $\phi^n(k)$ and $\phi^n(l)$ infer, with 2, that $\phi^n(k) \equiv \phi^n(l)$ and hence, by definition, $atom^n(k) = atom^n(l)$.

6: As $\phi^n(k)$ is irreducible, use 1 to prove that $k \Vdash \phi$ implies $\phi^n(k) \vdash \phi$. On the other hand, as $k \Vdash \phi^n(k)$, from $\phi^n(k) \vdash \phi$ infer that $\phi \in Th^n(k)$. ⊣

2.3.0.4. COROLLARY. *Every formula in $\mathbf{CpL}^n$ is equivalent to a disjunction of irreducible formulas in $\mathbf{Cpl}^n$.*

Proof. Obvious, as $Diag(\mathbf{CpL}^n)$ is a Boolean algebra with the irreducible formulas as its atoms. ⊣

2.3.0.5. THEOREM. *Let A^n be the set of types (irreducible formulas) of $\mathbf{CpL}^n$. Then A^n is an exact model of $\mathbf{CpL}^n$.*

Proof. As every formula in $\mathbf{CpL}^n$ is equivalent to a disjunction of irreducible formulas, according to corollary 2.3.0.4, there is a unique correspondence between the subsets of A^n and the equivalence classes of $\mathbf{CpL}^n$. ⊣

According to lemma 2.3.0.3, the type of a world k in an n-model is determined by the set $atom^n(k)$.

2.3.0.6. DEFINITION. *Let k be a node in a $\mathbf{CpL}$ model. Then $\tau^n(k)$, the* semantic type *of k in $\mathbf{CpL}^n$ is defined by*

$$\tau^n(k) = \langle atom^n(k), \emptyset \rangle.$$

The following fact justifies our choice for the definition of semantic type in $\mathbf{CpL}^n$. It is a simple consequence of the definition of semantic type in $\mathbf{CpL}^n$ and lemma 2.3.0.3.

2.3.0.7. FACT. *Let k and l be nodes in* **CpL** *models. Then*

$$\tau^n(k) = \tau^n(l) \quad \Leftrightarrow \quad \phi^n(k) \equiv \phi^n(l) \quad \Leftrightarrow \quad Th^n(k) = Th^n(l).$$

For K a **CpL** model, let K^τ be the set of n-types in K. K^τ may be treated as a **CpL** model, with $atom^n(\tau^n(k)) = atom^n(k)$. According to the facts above the models K and K^τ force the same $\mathbf{CpL}^n$ formulas. Of course the application of this reduction to K^τ would yield K^τ itself and hence we call K^τ *n-irreducible.*
Let $Exm(\mathbf{CpL}^n)$ be the set of all $\mathbf{CpL}^n$ types, i.e.

$$Exm(\mathbf{CpL}^n) = \{\langle Q, \emptyset\rangle \mid Q \subseteq \{p_1, \ldots, p_n\}\}.$$

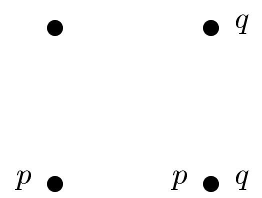
To make $Exm(\mathbf{CpL}^n)$ into a $\mathbf{CpL}^n$ Kripke model, use j_1 as the $atom^n$. If we use k_Q to denote the world in $Exm(\mathbf{CpL}^n)$ corresponding to the type $\langle Q, \emptyset\rangle$, then obviously $atom^n(k_Q) = Q$.

Clearly every k_Q corresponds to the type ϕ^n_Q defined above. Note that all subsets of $Exm(\mathbf{CpL}^n)$ are *closed*, as the accessibility relation is empty. Every subset $X \subseteq Exm(\mathbf{CpL}^n)$ corresponds to the disjunction of the ϕ^n_Q such that $k_Q \in X$, which proves that $Exm(\mathbf{CpL}^n)$ is an *exact Kripke model* of $\mathbf{CpL}^n$.

The following facts summarize these conclusions.

2.3.0.8. FACTS. *Let $Exm(\mathbf{CpL}^n)$ be the model defined above.*

1. *$Exm(\mathbf{CpL}^n)$ is (isomorphic to) the exact model of $\mathbf{CpL}^n$;*
2. *$Exm(\mathbf{CpL}^n)$ is an exact Kripke model of $\mathbf{CpL}^n$;*
3. *if K a* **CpL** *n-model that is an exact Kripke model of $\mathbf{CpL}^n$, then K is isomorphic to $Exm(\mathbf{CpL}^n)$;*
4. *$Exm(\mathbf{CpL}^n)$ has 2^n nodes and the Lindenbaum algebra of $\mathbf{CpL}^n$ has 2^{2^n} equivalence classes.*



3. FIGURE. *The exact Kripke model of $\mathbf{CpL}^2$.*

If K has the same set of worlds as $Exm(\mathbf{CpL}^n)$, but a non-empty accessibility relation (hence K is not a **CpL** Kripke model), then K is a universal model for $\mathbf{CpL}^n$. As every subset of nodes in K corresponds uniquely to an equivalence class in $\mathbf{CpL}^n$, K is also an exact Kripke model (where the set of 'closed subsets' in the definition of exact model is taken to be the set of all subsets[4]). Hence there are (up to isomorphism) $2^{2^{2n}}$ exact Kripke models of $\mathbf{CpL}^n$.

Note that $Exm(\mathbf{CpL}^n)$ would not have been a model if we had restricted the definition of a $\mathbf{CpL}^n$ model to single worlds k (or singleton sets), as is usual.

2.4 Types in modal logic

In this section we will introduce fragments of modal logic with restricted nesting of the box operator. Our logical framework will be the system $\mathbf{K}$, the rules and axioms of which were given in the general preliminary section of the introduction.

Recall the standard definition of *modal depth* of a formula, also known as *modal degree*, which we here prefer to call the level of *box nesting* in analogy with the level of nesting of the implication in $\mathbf{IpL}$ that will be used later on.

2.4.0.1. Definition. *The level of* box nesting *of a* $\mathbf{K}$ *formula is denoted by the inductively defined function* $\beta(\phi)$*:*

p *atom*: $\beta(p) = 0$;
$\phi = \psi \circ \chi$: $\beta(\phi) = max\{\beta(\psi), \beta(\chi)\}$ *if* $\circ \in \{\wedge, \vee, \rightarrow\}$;
$\phi = \neg\psi$: $\beta(\phi) = \beta(\psi)$;
$\phi = \Box\psi$: $\beta(\phi) = \beta(\psi) + 1$.

The fragment $\mathbf{K}^n_m$ will be the fragment with $\{p_1, \ldots, p_n\}$ as its set of propositional variables and the nesting of the box operator restricted by the condition $\beta(\phi) \leq m$.

2.4.0.2. Fact. *The Lindenbaum algebra of* $\mathbf{K}^n_m$ *is a finite Boolean algebra and the Lindenbaum algebra of* $\mathbf{K}^n$ *is an infinite Boolean algebra.*

If L is an extension of $\mathbf{K}$, that is, if L can be derived by adding axioms to $\mathbf{K}$ and L^n_m is defined like $\mathbf{K}^n_m$ above, the above fact is also true for L.

As finite Boolean algebras are *atomic*, both the diagrams of $\mathbf{K}^n_m$ and of L^n_m are generated by their atoms. As in the case of $\mathbf{CpL}$, treated in the previous section, these atoms can be proved to be *irreducible* (see definition 2.1.0.1).

Clearly the set of irreducible formulas (or their equivalence classes to be precise) in $\mathbf{K}^n_m$ is an exact model according to definition 1.2.0.8. Every formula in $\mathbf{K}^n_m$ is equivalent to a disjunction of irreducible formulas and in this way we have a 1–1 correspondence between formulas and sets of irreducible formulas.

The Lindenbaum algebra of $\mathbf{K}^n$ is also an atomic Boolean algebra, but this is not the case for the L^n of arbitrary extensions L of $\mathbf{K}$ (see [Bellissima 84]). The set of irreducible formulas in $\mathbf{K}^n$ is not an exact model, as there are infinite sets of irreducibles that do not correspond to a formula in $\mathbf{K}^n$.

Define $Th^n_m(k) = \{\psi \in \mathbf{K}^n_m \mid k \Vdash \psi\}$. From the fact that $\mathbf{K}^n_m$ is finite, we may conclude that $Th^n_m(k)$ is a finite theory and hence we can define a formula $\phi^n_m(k)$ in $\mathbf{K}^n_m$ as $\phi^n_m(k) = \bigwedge Th^n_m(k)$.

[4]Recall that in classical (modal) Kripke models the order of the general model of definition 1.2.0.8 has nothing to do with the accessibility relation in the Kripke model.

This formula $\phi^n_m(k)$ will be recognized as the *type* of k in $\mathbf{K}^n_m$ and for every l in a Kripke model L we have $l \Vdash \phi^n_m(k)$ iff $Th^n_m(k) = Th^n_m(l)$.

We will give a more explicit definition of the types of $\mathbf{K}^n_m$ in the sequel. First we try to find the semantic types in $\mathbf{K}^n_m$ and a characterization of the exact Kripke model of $\mathbf{K}^n_m$. To do so we will rephrase a theorem in [Fine 74] using the layered bisimulations introduced in the general preliminaries.

2.4.0.3. DEFINITION. *Nodes k and l in Kripke models are called n,m-*equivalent*, $k \equiv^n_m l$, if for all $\phi \in \mathbf{K}^n_m$*

$$k \Vdash \phi \quad \Leftrightarrow \quad l \Vdash \phi$$

(and hence $Th^n_m(k) = Th^n_m(l)$).

2.4.0.4. THEOREM. *Nodes k and l, in Kripke models K and L respectively, are n,m-equivalent iff $k \rightleftarrows^n_m l$.*

Proof. By induction on m. For $m = 0$ note that $k \rightleftarrows^n_0 l$ iff $atom^n(k) = atom^n(l)$, and that $Th^n_0(k)$ is the set of $\mathbf{CpL}^n$ formulas forced by k. Hence also: $Th^n_0(k) = Th^n_0(l) \quad \Leftrightarrow \quad atom^n(k) = atom^n(l)$.

Now assume the theorem proved for m. Let $k \rightleftarrows^n_{m+1} l$. We will prove $k \Vdash \phi$ to be equivalent with $l \Vdash \phi$ for all $\phi \in \mathbf{K}^n_{m+1}$ by showing $k \Vdash \phi$ implies $l \Vdash \phi$. We use induction on the length of ϕ.

The cases in which ϕ is atomic, a conjunction or a negation are obvious. So let $\phi = \Box\psi$ and $k \Vdash \Box\psi$. Note that as $\phi \in \mathbf{K}^n_{m+1}$ we know that $\psi \in \mathbf{K}^n_m$. Let $l_1 \in L$ be such that lRl_1. From $k \rightleftarrows^n_{m+1} l$ we infer that there is a $k_1 \in K$ such that kRk_1 and $k_1 \rightleftarrows^n_m l_1$. As $k_1 \Vdash \psi$, by our first induction hypothesis also $l_1 \Vdash \psi$. Which proves $l \Vdash \Box\psi$.

For the other direction, assume $k \equiv^n_{m+1} l$ and kRk_1. We have to prove the existence of an $l_1 \breve{R} l$ such that $l_1 \rightleftarrows^n_m k_1$, which, according to our induction hypothesis, is equivalent to $l_1 \equiv^n_m k_1$.

Now let $\phi^n_m(k_1)$ be the type of k_1 in $\mathbf{K}^n_m$ (as pointed out above, $\phi^n_m(k_1) = \bigwedge Th^n_m(k_1)$). As $k \Vdash \Diamond\phi^n_m(k_1)$ and $k \equiv^n_{m+1} l$ we will have also $l \Vdash \Diamond\phi^n_m(k_1)$ and for some $l_1 \breve{R} l$ it must be true that $l_1 \Vdash \phi^n_m(k_1)$. As observed above this implies that $k_1 \equiv^n_m l_1$.

By interchanging the rôles of k and l the n,m-bisimulation condition in the other direction is proved in the same way. $\dashv$

In [Fine 74] Fine only proved one direction of this theorem, i.e.

$$k \rightleftarrows^n_m l \quad \Rightarrow \quad Th^n_m(k) = Th^n_m(l).$$

Fine did not use layered bisimulation, but *m-equivalence*, a notion that is easily proved equivalent with our notion of n,m-bisimulation[5]. The analogy with results of

[5]That is k and l are m-equivalent according to the definition in [Fine 74], iff k and l n,m-bisimulate each other.

Fraïssé and Ehrenfeucht for first order theories, that was mentioned in Fine's article, will be taken up in the last section of this chapter.

As a simple corollary of theorem 2.4.0.4, for each n and m $k \rightleftarrows^n_m l$ will be equivalent to $k \equiv^n_m l$ (or $Th^n_m(k) = Th^n_m(l)$). In general $\forall m(k \equiv^n_m l)$ does not imply $k \rightleftarrows^n l$, as example 2.2.0.4 provides us with a counter-example.

The semantic types that we will define for $\mathbf{K}^n_m$ are quite natural characterizations of the equivalence classes of the n, m-bisimulations.

2.4.0.5. DEFINITION. *Let k be a node in a finite n-model. Then the* semantic n, m-type *of k (in $\mathbf{K}$), $\tau^n_m(k)$, is defined by:*

- $\tau^n_0(k) = \langle atom^n(k), \emptyset \rangle$;
- $\tau^n_{m+1}(k) = \langle atom^n(k), \{ \tau^n_m(l) \mid kRl \} \rangle$.

The set of all semantic n, m-types $\tau^n_m(k)$ is written T^n_m.

This definition is justified by the following lemma.

2.4.0.6. LEMMA. *If k, l are nodes in finite Kripke models then*

$$\tau^n_m(k) = \tau^n_m(l) \quad \Leftrightarrow \quad k \equiv^n_m l.$$

Proof. We will apply theorem 2.4.0.4 and prove $\tau^n_m(k) = \tau^n_m(l) \quad \Leftrightarrow \quad k \rightleftarrows^n_m l$. We will proceed by introducing a relation $\sim^n_m$ between K and L, defined as $k \sim^n_m l \;\Leftrightarrow\; \tau^n_m(k) = \tau^n_m(l)$, and prove $\sim^n_m$ to be an n, m-bisimulation, using induction on m.

By the definition of the semantic n, m-types, $k \sim^n_m l$ implies $atom^n(k) = atom^n(l)$. This proves the case that $m = 0$ and the first condition for an n, m-bisimulation in general. To prove the other conditions for an $n, m+1$-bisimulation, assume $k \sim^n_{m+1} l$ and kRk'. From $\tau^n_{m+1}(k) = \tau^n_{m+1}(l)$ it follows that $\tau^n_m(k') \in j_1(\tau^n_{m+1}(l))$ and hence there is an $l'\breve{R}l$ such that $\tau^n_m(k') = \tau^n_m(l')$. As by definition $\tau^n_m(k') = \tau^n_m(l') \quad \Leftrightarrow \quad k' \sim^n_m l$, this proves the first condition of n, m-bisimulation. The second condition is proved in the same way, interchanging the rôles of k and l and k' and l'.

For the proof of the other direction we will also use induction on m. The case $m = 0$ is again trivial, so suppose $k \rightleftarrows^n_{m+1} l$. Then obviously it will be true that $atom^n(k) = atom^n(l)$. To prove that also $j_1(\tau^n_{m+1}(k)) = j_1(\tau^n_{m+1}(l))$, let kRk'. By the definition of $\rightleftarrows^n_{m+1}$ there should be an $l'\breve{R}l$ such that $k' \rightleftarrows^n_m l'$. Using the induction hypothesis, it follows that $\tau^n_m(k') = \tau^n_m(l')$. Which proves $j_1(\tau^n_{m+1}(k)) \subseteq j_1(\tau^n_{m+1}(l))$. For the other direction of the inclusion, interchange the rôles of k and l. $\dashv$

In [Bellissima 84], $\phi^n_m(k)$, the $\mathbf{K}^n_m$ *type* of k (n, m-types for short), is defined as follows.[6] (Recall definition 2.3.0.1 for $\phi^n_{\mathbf{CpL}}(k)$).

[6]This definition is essentially the same as that of an $m, \vec{p}$-type in [Shavrukov 93].

2.4.0.7. DEFINITION. *Let k be a node in a finite n-model. Define $\phi^n_m(k)$, the $\mathbf{K}^n_m$* type *of k inductively as:*

- $\phi^n_0(k) = \phi^n_{\mathbf{CpL}}(k)$;
- $\phi^n_{m+1}(k) = \phi^n_{\mathbf{CpL}}(k) \wedge \bigwedge\{\Diamond\phi^n_m(l) \mid kRl\} \wedge \Box\bigvee\{\phi^n_m(l) \mid kRl\}$.

Let A^n_m be the set of (equivalence classes of) n,m-types.

2.4.0.8. FACT. *If k is a a node in a finite n-model, then for all m*

$$k \Vdash \phi^n_m(k).$$

The proof of this fact is obvious.

The next lemma shows that $\phi^n_m(k)$ indeed is an axiom for $Th^n_m(k)$ (using theorem 2.4.0.6) and hence is a type.

2.4.0.9. LEMMA. *Let k and l be nodes in finite n-models. If $k \Vdash \phi^n_m(l)$, then $\tau^n_m(k) = \tau^n_m(l)$.*

Proof. We will use induction on m. If $m = 0$, then $k \Vdash \phi^n_{\mathbf{CpL}}(l)$ implies $atom^n(k) = atom^n(l)$ and hence $\tau^n_0(k) = \tau^n_0(l)$. So assume $k \Vdash \phi^n_{m+1}(l)$. Then $k \Vdash \phi^n_{\mathbf{CpL}}(l)$ and we may infer that $atom^n(k) = atom^n(l)$. To prove that also $j_1(\tau^n_{m+1}(k)) = j_1(\tau^n_{m+1}(l))$, we show $j_1(\tau^n_{m+1}(k)) \subseteq j_1(\tau^n_{m+1}(l))$.

Let kRr and hence $\tau^n_m(r) \in j_1(\tau^n_{m+1}(k))$. From $k \Vdash \phi^n_{m+1}(l)$, by definition 2.4.0.7, infer that $r \Vdash \bigvee\{\phi^n_m(s) \mid lRs\}$. So, for some $s\breve{R}l$ we have $r \Vdash \phi^n_m(s)$ and, by the induction hypothesis, $\tau^n_m(r) = \tau^n_m(s)$. Which proves $\tau^n_m(r) \in j_1(\tau^n_{m+1}(l))$.

To prove $j_1(\tau^n_{m+1}(l)) \subseteq j_1(\tau^n_{m+1}(k))$, let lRs and hence $\tau^n_m(s) \in j_1(\tau^n_{m+1}(l))$. As $k \Vdash \phi^n_{m+1}(l)$, by definition 2.4.0.7, we may infer that $k \Vdash \Diamond\phi^n_m(s)$. Hence, for some $r\breve{R}k$, $r \Vdash \phi^n_m(s)$. By the induction hypothesis, this implies $\tau^n_m(r) = \tau^n_m(s)$, which proves $\tau^n(s) \in j_1(\tau^n_{m+1}(k))$. $\dashv$

The lemmas 2.4.0.6 and 2.4.0.9 combine into:

2.4.0.10. THEOREM. *The set T^n_m of semantic n,m-types corresponds exactly to the set A^n_m of types in $\mathbf{K}^n_m$, in the sense that:*

$$\forall l \in K(\, l \Vdash \phi^n_m(k) \quad \Leftrightarrow \quad \tau^n_m(l) = \tau^n_m(k)).$$

Lemma 2.4.0.10 immediately leads to:

2.4.0.11. COROLLARY. *If K is a Kripke model such that each n,m-type occurs exactly once in K, then the subsets of K correspond exactly to the equivalence classes of $\mathbf{K}^n_m$. That is, the following function $[\![.]\!]$ is an isomorphism:*

$$[\![\phi]\!] = \{k \in K \mid k \Vdash \phi\}.$$

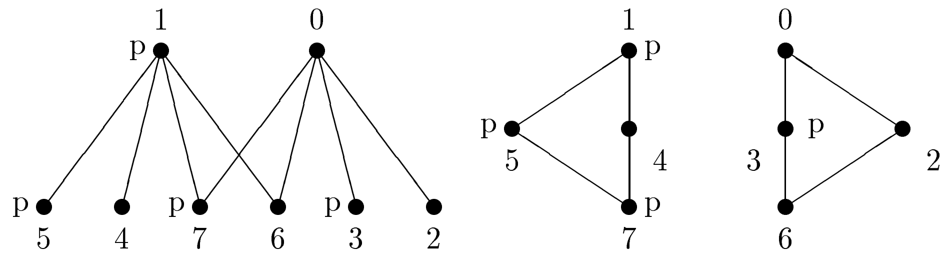
It can be proved that such an *exact model*, in which each subset corresponds to a formula and vice versa, exists for each $\mathbf{K}^n_m$.

2.4.0.12. THEOREM. *For each n and m there exists an exact Kripke-model K for $\mathbf{K}^n_m$, i.e., for each $U \subseteq K$, there is a formula ϕ in L^n_m such that $\{k \in K \mid k \Vdash \phi\} = U$, and K is n,m-complete, in the sense that for all $\phi, \psi \in L^n_m$, $\{k \in K \mid k \Vdash \phi\} = \{k \in K \mid k \Vdash \psi\}$ iff $\vdash \phi \leftrightarrow \psi$.*

Proof. We apply the so-called Henkin method to the (up to equivalence) finite set of formulas in $\mathbf{K}^n_m$, which is closed under taking subformulas. This gives one a Kripke-model with the maximal consistent sets as its worlds, with $\Gamma\, R\, \Delta$ defined by: for each $\Box\gamma \in \Gamma$, γ is an element of Δ and $\Gamma \Vdash p_i$ by: $p_i \in \Gamma$. The maximal consistent sets can be replaced by their conjunctions which are exactly the irreducible elements of $\mathbf{K}^n_m$. So a subset of the model will correspond to a disjunction of irreducibles, i.e. an arbitrary formula of $\mathbf{K}^n_m$. Obviously, non-equivalent formulas are forced on different subsets of the model. ⊣

The Henkin construction above also works for fragments L^n_m, where L is an extension of $\mathbf{K}$. The result in each case is called the *canonical exact model* . But the frame of the resulting model is not necessarily a frame for the logic L.

Unlike the exact models of fragments of intuitionistic propositional logic (see [JHR 91], [Hendriks 93]), not all the exact models of L^n_m are necessarily isomorphic.



4. FIGURE. *Two exact models for $\mathbf{K}^1_1$.*

The formulas in the exact models of $\mathbf{K}^1_1$:

0. $\neg p \wedge \Box\bot$	4. $\neg p \wedge \Diamond p \wedge \Box p$
1. $p \wedge \Box\bot$	5. $p \wedge \Diamond p \wedge \Box p$
2. $\neg p \wedge \Diamond\neg p \wedge \Box\neg p$	6. $\neg p \wedge \Diamond p \wedge \Diamond\neg p$
3. $p \wedge \Diamond\neg p \wedge \Box\neg p$	7. $p \wedge \Diamond p \wedge \Diamond\neg p$

The accessibility relation defined in a canonical exact model corresponds to the relation between irreducible elements α and β of L^n_m defined as:

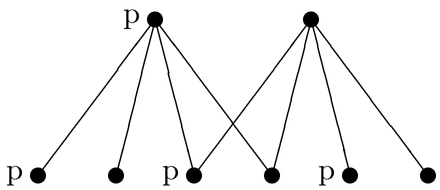
$$\alpha R \beta \quad \Leftrightarrow \quad \alpha, \Diamond\beta \nvdash \bot.$$

It is often possible to restrict this relation. For example in such a way that the Kripke exact model belongs to a certain subclass of the class of Kripke models (the reflexive models, well-founded models and so on). Note that in this way the completeness theorems for $\mathbf{K}$ and some of its extensions can be proved (see for example [HC 84]).

For the infinite fragment $\mathbf{K}^n$ there is no exact model in which all subsets determine a formula, but there is a (infinite) model which is n-complete. We will give the construction of such a model and call it ExK^n. Our ExK^n is comparable to the n-complete model given in [Grigolia 83] and [Rybakov 89] for provability logic[7].

2.4.0.13. DEFINITION. *ExK^n with its R and $\Vdash$ is defined as the union of inductively defined ExK^n_m for $m \in \omega$.*

- $ExK^n_0 = \mathcal{P}(\{p_1, \cdots, p_n\})$, *the elements of* ExK^n_0 *are all* R-*incomparable, and* $Q \Vdash p \;\Leftrightarrow\; p \in Q$;
- $ExK^n_{m+1} = \{\langle Q, X\rangle \mid Q \subseteq \{p_1, \cdots, p_n\}, X \subseteq \bigcup_{i\leq m} ExK^n_i, X \cap ExK^n_m \neq \emptyset\}$, $\langle Q, X\rangle R\, Y \;\Leftrightarrow\; Y \in X$, *and* $\langle Q, X\rangle \Vdash p \;\Leftrightarrow\; p \in Q$;
- $ExK^n = \bigcup_{i\in\omega} ExK^n_i$.



5. FIGURE. *The model $ExK^1_0 \cup ExK^1_1$.*

From this picture it can be calculated that ExK^1_2 will have 504 nodes.
It is obvious from the construction that each n, m-type will be realized by some $k \in ExK^n$. This ensures the n-completeness of ExK^n.

The above definition is such that each node in ExK^n_i is the root of a finite reverse well-founded (and hence irreflexive) Kripke model.

2.4.0.14. FACT. **K** *is complete for finite reverse well-founded Kripke models.*

Let us use the completeness of **K** for finite and reverse well-founded Kripke models to define semantic types in **K**.

2.4.0.15. DEFINITION. *For a node k in a finite reverse well-founded Kripke model define the semantic type in* **K**:

$$\tau^n(k) = \langle atom^n(k), \{\tau^n(l) \mid kRl\}\rangle.$$

As we are dealing with finite reverse well-founded models we may use $\delta(k)$, the depth of node k, to show that definition 2.4.0.15 is sound. Obviously $\tau^n(k) \notin j_1(\tau^n(k))$.

Observe that in ExK^n all semantic n-types of nodes in finite, reverse well-founded Kripke models are realized.

The definition of semantic types for $\mathbf{K}^n$ would not be of any use without the following lemma, stating its relation with bisimulation.

[7] Similar constructions for fragments in **K4Grz** and **S4** may be found in [Shehtman 78].

2.4.0.16. LEMMA. *If k and l are nodes in finite, irreflexive and reverse well-founded models, then:*

$$\tau^n(k) = \tau^n(l) \quad \Leftrightarrow \quad k \rightleftarrows^n l.$$

Proof. Define $d = \max\{\delta(k), \delta(l)\}$. We will proceed by induction on d. In case $d = 0$, both k and l are terminal nodes, and then the lemma is trivial as both sides of the equivalence sign are equivalent to $atom^n(k) = atom^n(l)$.

So suppose $d > 0$. If $k \rightleftarrows^n l$ then trivially $atom^n(k) = atom^n(l)$. To prove $j_1(\tau^n(k)) \subseteq j_1(\tau^n(l))$, assume kRk'. As k and l bisimulate each other there is an $l'\breve{R}l$ such that $k' \rightleftarrows l'$. The maximum of the depth of k' and l' is less than d and hence, by the induction hypothesis, $\tau^n(k') = \tau^n(l')$. So $\tau^n(k') \in j_1(\tau^n(l))$, which proves $j_1(\tau^n(k)) \subseteq j_1(\tau^n(l))$. As the proof of $j_1(\tau^n(l)) \subseteq j_1(\tau^n(k))$ is similar, we may conclude that $\tau^n(k) = \tau^n(l)$.

If $\tau^n(k) = \tau^n(l)$ then again trivially $atom^n(k) = atom^n(l)$. Assume kRk'. Then $\tau^n(k') \in j_1(\tau^n(l))$ and so there is an $l'\breve{R}l$ such that $\tau^n(l') = \tau^n(k')$. By applying the induction hypothesis infer that $k' \rightleftarrows^n l'$. As the other condition for bisimulation is proved likewise, we conclude that $k \rightleftarrows^n l$. $\dashv$

In general, to be able to construct a universal model (a *minimal* complete model) from the semantic types of nodes in M-models, there should not be too many models in M.

For example, $\mathbf{K}^n$ is complete for the class of finite n-models, but also for the subclass of finite reverse well-founded n-models. Assume that we would have defined a semantic type $\tau^n(k)$ for nodes k in finite n-models (in general), such that lemma 2.4.0.16 holds. Then clearly the set of these new semantic types would contain too many semantic types to be the universe of a *minimal* complete model for $\mathbf{K}^n$.

To prove that ExK^n is a universal model for $\mathbf{K}^n$ (and hence that the class of finite reverse well-founded models is small enough) we will define a type $\phi^n(k)$ (in $\mathbf{K}^n$) for every node in ExK^n, in such a way that $[\![\phi^n(k)]\!] = \{k\}$.

The definition of these types seems to belong to modal logic folklore (see for example [Bellissima 84]) and is very similar to the definition of the n,m-types above.

2.4.0.17. DEFINITION. *Let k be a node in a finite reverse well-founded Kripke model. Define $\phi^n(k)$, the* type *of k in $\mathbf{K}^n$, by:*

$$\phi^n(k) = \phi^n_{\mathbf{CpL}}(k) \wedge \{\Diamond\phi(l) \mid kRl\} \wedge \Box\bigvee\{\phi(l) \mid kRl\}.$$

Define $\bigwedge\emptyset = \bot$ and $\bigvee\emptyset = \top$, and observe that if k is a terminal node, $\phi^n(k) = \phi^n_{\mathbf{CpL}} \wedge \Box\bot$. That the types we defined for $\mathbf{K}^n$ correspond exactly to the semantic types of $\mathbf{K}^n$ is a corollary of the next theorem.

2.4.0.18. LEMMA. *If k and l are nodes in finite reverse well-founded Kripke models, then $k \Vdash \phi^n(l)$ implies $\tau^n(k) = \tau^n(l)$.*

Proof. We will use lemma 2.4.0.16 and prove $k \Vdash \phi^n(l)$ implies $k \rightleftarrows^n l$. We will use induction on $\delta(k)$, the depth of k. Note that, as $\phi^n(l)$ implies $\phi^n_{\mathbf{CpL}}(l)$, we may infer that $atom^n(k) = atom^n(l)$. Now assume $k \Vdash \phi^n(l)$. In case $\delta(k) = 0$,we know that k is a terminal node and $k \Vdash \Box\bot$. Note that l will also be a terminal node. For lRl' would imply $k \Vdash \Diamond\phi^n(l')$ which would make $Th^n(k)$ inconsistent. For terminal nodes $atom^n(k) = atom^n(l)$ implies $k \rightleftarrows^n l$.

So let $\delta(k) > 0$. If $k'\breve{R}k$, then $\phi^n(l) \nVdash \Box\bot$ and hence $k' \Vdash \bigvee\phi^n(l_i)$ (where the l_i are the successors of l). Hence, for some $l'\breve{R}l$, $k' \Vdash \phi^n(l')$. Using the induction hypothesis we may conclude that $k' \rightleftarrows^n l'$.

Now let lRl'. Then $k \Vdash \Diamond\phi^n(l')$ and hence for some $k'\breve{R}k$ we have $k' \Vdash \phi^n(l')$. Again by the induction hypothesis we conclude $k' \rightleftarrows^n l'$. Which proves $k \rightleftarrows^n l$. ⊣

2.4.0.19. THEOREM. *If k and l are nodes in finite reverse well-founded Kripke models, then*

$$k \Vdash \phi^n(l) \quad \Leftrightarrow \quad \tau^n(k) = \tau^n(l) \quad \Leftrightarrow \quad Th^n(k) = Th^n(l).$$

Proof. By lemma 2.4.0.16 $\tau^n(k) = \tau^n(l)$ is equivalent with $k \rightleftarrows^n l$ and (by the bisimulation theorem) hence implies $Th^n(k) = Th^n(l)$. As $\phi^n(l) \in Th^n(l)$, from $Th^n(k) = Th^n(l)$ we may infer that $k \Vdash \phi^n(l)$. On the other hand, by lemma 2.4.0.18, $k \Vdash \phi^n(l)$ implies $\tau^n(k) = \tau^n(l)$. ⊣

2.5 Types and reductions in IpL

In the semantics of **IpL** we confine our attention mainly to finite, transitive, reflexive and anti-symmetric Kripke models (the finite **IpL** models).

2.5.0.1. DEFINITION. *Let k be a node in a finite* **IpL** *model. The* semantic type *of k in* **IpL**, *$\tau^n(k)$, is defined by induction on $\delta(k)$, the depth of k.*

$$\tau^n(k) = \langle atom^n(k), \{\tau^n(l) \mid k < l \text{ and if } atom^n(k) = atom^n(l) \\ \text{then } \exists k' > k(\tau^n(k') \neq \tau^n(l) \wedge \tau^n(k') \notin j_1(\tau^n(l)))\}\rangle.$$

Define the order of semantic types in **IpL** *as:*

$$t \preceq t' \quad \Leftrightarrow \quad t = t' \text{ or } t' \in j_1(t).$$

Observe that, as a special case of this definition, we have $\tau^n(k) = \langle atom^n(k), \emptyset\rangle$ if $\delta(k) = 0$. Definition 2.5.0.1 is rather complex in comparison to definition 2.4.0.15, due to the fact that the accessibility relation is reflexive in this case.

2.5.0.2. LEMMA. *Let k and l be nodes in a finite* **IpL** *model and $k < l$. Then*

1. *if $atom^n(k) \neq atom^n(l)$ then $\tau^n(l) \in j_1(\tau^n(k))$;*
2. *$j_1(\tau^n(l)) \subseteq j_1(\tau^n(k))$;*
3. *$\tau^n(k) \neq \tau^n(l) \quad \Leftrightarrow \quad \tau^n(l) \in j_1(\tau^n(k))$;*
4. *$\tau^n(k) \preceq \tau^n(l)$.*

Proof. 1: This is a simple consequence of definition 2.5.0.1.

2: Let $l < l'$ in such a way that $\tau^n(l') \in j_1(\tau^n(l))$. As obviously $k < l'$, if $atom^n(k) \neq atom^n(l')$ then $\tau^n(l') \in j_1(\tau^n(k))$. Now suppose that $atom^n(k) = atom^n(l')$. From $k < l < l'$ infer that $atom^n(l) = atom^n(l')$. From definition 2.5.0.1, infer that $\exists k' > l(\tau^n(k') \neq \tau^n(l') \wedge \tau^n(k') \notin j_1(\tau^n(l')))$. As $k < l$ also $\exists k' > k(\tau^n(k') \neq \tau^n(l') \wedge \tau^n(k') \notin j_1(\tau^n(l')))$ and from definition 2.5.0.1 infer that $\tau^n(l') \in j_1(\tau^n(k))$, which proves $j_1(\tau^n(l)) \subseteq j_1(\tau^n(k))$.

3: Obviously $\tau^n(k) \notin j_1(\tau^n(k))$, from which the $\Leftarrow$ part follows trivially.

To prove the $\Rightarrow$ part, suppose that $\tau^n(l) \notin j_1(\tau^n(k))$. From the first part of the lemma we may conclude that $atom^n(k) = atom^n(l)$. According to definition 2.5.0.1, for every $k' > k$ it will be the case that $\tau^n(k') = \tau^n(l)$ or $\tau^n(k') \in j_1(\tau^n(l))$. Hence, if $k' > k$ and $\tau^n(k') \in j_1(\tau^n(k))$ then, by the assumption that $\tau^n(l) \notin j_1(\tau^n(k))$, $\tau^n(k') \neq \tau^n(l)$ and so we may conclude that $\tau^n(k') \in j_1(\tau^n(l))$. Which proves $j_1(\tau^n(k)) \subseteq j_1(\tau^n(l))$. In combination with the second part of the lemma, we conclude that $j_1(\tau^n(k)) = j_1(\tau^n(l))$ and hence $\tau^n(k) = \tau^n(l)$.

4: Observe that from $\tau^n(k) \neq \tau^n(l) \quad \Leftrightarrow \quad \tau^n(l) \in j_1(\tau^n(k))$ we may infer that $\tau^n(k) = \tau^n(l)$ or $\tau^n(l) \in j_1(\tau^n(k))$ and hence $\tau^n(k) \preceq \tau^n(l)$. $\dashv$

To prove that the semantic types introduced above do indeed satisfy the condition that $\tau^n(k) = \tau^n(l)$ implies $Th^n(k) = Th^n(l)$, we will use a theorem stating in effect that the semantic types are equivalence classes for n-bisimulation.

2.5.0.3. THEOREM. *For nodes k and l in finite* **IpL** *models, we have*

$$\tau^n(k) = \tau^n(l) \quad \Leftrightarrow \quad k \mathrel{\underline{\rightleftarrows}}^n l.$$

Proof. $\Rightarrow$: We will prove that the relation $k \sim^n l$, defined as $\tau^n(k) = \tau^n(l)$, is an n-bisimulation. It is trivial that the first condition for bisimulation, $atom^n(k) = atom^n(l)$, will apply. As the two remaining conditions are symmetric, we will prove only the first.

Suppose we know that $\tau^n(k) = \tau^n(l)$ and $k \leq k'$. We have to show that there is an $l' \geq l$ such that $\tau^n(k') = \tau^n(l')$. In case we have $\tau^n(k') = \tau^n(k)$ of course $l' = l$ will do. So assume that $\tau^n(k') \neq \tau^n(k)$. Using lemma 2.5.0.2, infer that $\tau^n(k') \in j_1(\tau^n(k))$ and hence, as $\tau^n(k) = \tau^n(l)$, $\tau^n(k') \in j_1(\tau^n(l))$. So, for some $l' > l$ we have $\tau^n(k') = \tau^n(l')$.

$\Leftarrow$: Let $k \mathrel{\underline{\rightleftarrows}}^n l$ and define $d = \max\{\delta(k), \delta(l)\}$. With induction on d we will prove $\tau^n(k) = \tau^n(l)$. Note that from $k \mathrel{\underline{\rightleftarrows}}^n l$ we may infer that $atom^n(k) = atom^n(l)$. We will prove $j_1(\tau^n(k)) \subseteq j_1(\tau^n(l))$. The proof of $j_1(\tau^n(l)) \subseteq j_1(\tau^n(k))$ is essentially the same, interchanging the rôles of k and l. As $atom^n(k) = atom^n(l)$, we may conclude that $\tau^n(k) = \tau^n(l)$.

Suppose that $k < k_1$ and $\tau^n(k_1) \in j_1(\tau^n(k))$. As $k \mathrel{\underline{\rightleftarrows}}^n l$, there is a $l_1 \geq l$ with $k_1 \mathrel{\underline{\rightleftarrows}}^n l_1$. Assume that $\tau^n(l_1) = \tau^n(l)$. Using the, already proved, first part of the theorem, then $l \mathrel{\underline{\rightleftarrows}}^n l_1$ and hence also $k_1 \mathrel{\underline{\rightleftarrows}}^n k$. From $k_1 \mathrel{\underline{\rightleftarrows}}^n k$ we may infer that $atom^n(k_1) = atom^n(k)$. As $\tau^n(k_1) \in j_1(\tau^n(k))$, there is, according to definition 2.5.0.1, a $k_2 > k$ such that $\tau^n(k_2) \neq \tau^n(k_1)$ and $\tau^n(k_2) \notin j_1(\tau^n(k_1))$. The fact that $k_1 \mathrel{\underline{\rightleftarrows}}^n k$ implies that there is a $k_3 \geq k_1$ with $k_3 \mathrel{\underline{\rightleftarrows}}^n k_2$. Note that both

$k < k_2$ and $k < k_3$. Hence by the induction hypothesis we have $\tau^n(k_2) = \tau^n(k_3)$. So, $\tau^n(k_3) \neq \tau^n(k_1)$ and $\tau^n(k_3) \notin j_1(\tau^n(k_1))$, contradicting lemma reflemt4.3 applied to $k_1 \leq k_3$.

From this contradiction we infer that $\tau^n(l_1) \neq \tau^n(l)$. As $l \leq l_1$, again by lemma 2.5.0.2.3, we conclude that $\tau^n(l_1) \in j_1(\tau^n(l))$. Hence, we have $l < l_1$ and $k < k_1$. By the induction hypothesis we infer from $k_1 \rightleftarrows^n l_1$ that $\tau^n(k_1) = \tau^n(l_1)$ and so $\tau^n(k_1) \in j_1(\tau^n(l))$, what had to proved. $\dashv$

2.5.0.4. Corollary. *Let k be a node in a finite* **IpL** *model and $k < l$. Then*

$$\tau^n(k) = \langle atom^n(k), \{\tau^n(l) \mid k < l \wedge \neg(k \rightleftarrows^n l)\}\rangle.$$

Proof. We prove that for $k < l$ we have $\tau^n(l) \in j_1(\tau^n(k))$ iff $\neg(k \rightleftarrows^n l)$. By lemma 2.5.0.2.3, If $k < l$, then $\tau^n(l) \in j_1(\tau^n(k))$ is equivalent to $\tau^n(k) \neq \tau^n(l)$. Now apply theorem 2.5.0.3. $\dashv$

Let us write $Th^n(k)$ for the $\mathbf{IpL}^n$ theory of a node k in a finite **IpL** model. Hence, $Th^n(k) = \{\phi \in \mathbf{IpL}^n \mid k \Vdash \phi\}$.

2.5.0.5. Lemma. *Let k and l be nodes in finite* **IpL** *models. If $\tau^n(k) \preceq \tau^n(l)$ then $Th^n(k) \subseteq Th^n(l)$.*

Proof. Let $\tau^n(k) \preceq \tau^n(l)$. If $\tau^n(k) = \tau^n(l)$, then by the bisimulation theorem, theorem 2.2.0.2, $Th^n(k) = Th^n(l)$. On the other hand, if $\tau^n(k) \neq \tau^n(l)$ then there is a $k' > k$ such that $\tau^n(k') = \tau^n(l)$ and hence $Th^n(k') = Th^n(l)$. In an **IpL** model, from $k' > k$ infer $Th^n(k) \subseteq Th^n(k')$. $\dashv$

By ordering the semantic types in an **IpL** model K we will construct a new model, K^τ, a *maximal reduction*[8] of K.

2.5.0.6. Definition. *Let K be a finite* **IpL** *model. Define K^τ,* the maximal reduction *of K, by:*

$$K^\tau = \langle\{\tau^n(k) \mid k \in K\}, \preceq, j_0\rangle.$$

If K and K^τ are isomorphic, K is called an irreducible *model.*

The proofs of the following facts are straightforward.

2.5.0.7. Facts. *Let K be a finite* **IpL** *Kripke model.*

1. *K^τ is a Kripke model (note that $atom^n(\tau^n(k)) = j_0(\tau^n(k))$);*
2. *τ^n is a reduction from K to K^τ;*
3. *K^τ is irreducible.*

[8] The reader familiar with [Hendriks 93] will recognise the analogy with the γ-reduction introduced there.

As in modal logic, the semantic types in $\mathbf{IpL}^n$ correspond to formula types. The definition of the n-type of a node k in a finite ($\mathbf{IpL}$) Kripke model K is a result of a theorem of de Jongh (in [De Jongh 68], [De Jongh 70] and [JC 95]).

2.5.0.8. Definition. *Let k be a node in a finite irreducible* **IpL** *model. Define both $\theta^n(k)$ and $\chi^n(k)$ inductively over $\delta(k)$, the depth of k.*
Let

1. $Newatom^n(k) = \{q \in \{p_1, \ldots, p_n\} \mid k \nVdash q \text{ and } \forall l > k(l \Vdash q)\}$,
2. $\Psi^n(k) = \bigvee\{\chi^n(l) \mid k <_1 l\}$,
3. $\Phi^n(k) = \bigvee\{\theta^n(l) \mid k <_1 l\}$.

Then for

$\delta(k) = 0$: $\theta^n(k) = \phi^n_{CpL}(k)$; $\chi^n(k) = \neg\theta^n(k)$,
$\delta(k) > 0$: $\theta^n(k) = \bigwedge atom^n(k) \wedge (\bigvee Newatom^n(k) \vee \Psi^n(k) \to \Phi^n(k))$;
$\chi^n(k) = \theta^n(k) \to \Phi^n(k)$.

2.5.0.9. Theorem. (Jankov/De Jongh) *If k and l are nodes in irreducible finite* **IpL** *n-models then:*

1. $l \Vdash \theta^n(k) \quad \Leftrightarrow \quad k \leq l$,
2. $l \nVdash \chi^n(k) \quad \Leftrightarrow \quad l \leq k$.

Proof. We will prove 1 and 2 simultaneously by induction on the depth of k. In case $\delta(k) = 0$, both 1 and 2 are obvious. Assume the lemma for $\delta(k) \leq m$ and let $\delta(k) = m + 1$.

1. $\Rightarrow$: Let $l \Vdash \theta^n(k)$. If $l \Vdash \Phi^n(k)$ then for some h such that $k \leq_1 h$ we have $l \Vdash \theta^n(h)$. By the induction hypothesis this would imply $k \leq_1 h \leq l$ and hence $k \leq l$.

On the other hand, we will show that $l \nVdash \Phi^n(k)$ implies $k = l$. From $l \Vdash \theta^n(k)$ we may conclude that $l \Vdash \bigwedge atom^n(k)$. As $l \nVdash \Phi^n(k)$, we also may conclude $l \nVdash \bigvee Newatom^n(k)$ and $l \nVdash \Psi^n(k)$. By the induction hypothesis we infer that $l \leq h$ for all h such that $k \leq_1 h$. So if $q \in atom^n(l) \setminus atom^n(k)$, then we would have $q \in Newatom^n(k)$, contradicting $l \nVdash \bigvee Newatom^n(k)$. Hence $atom^n(l) = atom^n(k)$.

To prove that l also has the same successors as k, let g have a minimal depth such that $l \leq g$ and for all h with $k \leq_1 h$, $h \nleq h$. From the induction hypothesis it follows that $g \nVdash \Phi^n(k)$. As g is a successsor of l, we have also $g \Vdash \theta^n(k)$. In the same way as we proved for l, we may prove for g that $atom^n(g) = atom^n(k)$ and $g \leq h$ for all h such that $k \leq_1 h$. For g there is no proper successor g' which is not a successor of k. Otherwise g' would be a successor of l with $\delta(g') < \delta(g)$, $l \leq g$ and for all h with $k \leq_1 h$, $h \nleq h$, contradicting the minimality of (the depth) of g. From the irreducibility of the model conclude $g = k$ and hence $l \leq g$ implies $k \leq g$. Again by the irreducibility of the model infer $k = l$.

1. $\Leftarrow$: We first prove that $k \Vdash \theta^n(k)$. As obviously $k \Vdash \bigwedge atom^n(k)$ we still have to prove $k \Vdash \bigvee Newatom^n(k) \vee \Psi^n(k) \to \Phi^n(k)$. Observe that if $k <_1 h$ then by our induction hypothesis $k \nVdash \chi^n(h)$. Hence we may conclude that $k \nVdash \Psi^n(k)$. Of course, by the definition of $Newatom^n(k)$, also $k \nVdash \bigvee Newatom^n(k)$. So, if we assume $k \leq g$ with $g \Vdash \bigvee Newatom^n(k) \vee \Psi^n(k)$, then $k < g$. Hence, $g \Vdash \theta^n(h)$ for some h such

that $k <_1 h$ and hence $g \Vdash \Phi^n(k)$ So infer that $k \Vdash \theta^n(k)$ and apply lemma 2.5.0.5 to conclude that if $k \leq l$, then $l \Vdash \theta^n(k)$.

2. $\Rightarrow$: Assume that $l \nVdash \chi^n(k)$. Then for some g such that $l \leq g$, $g \Vdash \theta^n(k)$ and $g \nVdash \Phi^n(k)$. From the first part of this proof infer that $k \leq g$ and $h \nleq g$ for all h such that $k <_1 h$. Hence conclude that $= k$, which proves $l \leq k$.

2. $\Leftarrow$: As k is a node in an irreducible model, we have $\tau^n(k) \neq \tau^n(h)$ for all h such that $k <_1 h$ and, by induction hypothesis, $k \nVdash \theta^n(h)$. Hence, as $k \Vdash \theta^n(k)$ it should be true that $k \nVdash \Psi^n(k)$. A fortiori, for $l \leq k$ it is true that $l \nVdash \chi^n(k)$. ⊣

Recall that τ^n is a reduction from K to K^τ, as defined in 2.5.0.6.

2.5.0.10. DEFINITION. *For k a node in a finite* **IpL** *model define*

$$\phi^n(k) = \theta^n(\tau^n(k))$$

and

$$\psi^n(k) = \chi^n(\tau^n(k)).$$

2.5.0.11. THEOREM. *If k and l are nodes in finite* **IpL** *n-models then:*

1. $l \Vdash \phi^n(k) \quad \Leftrightarrow \quad \tau^n(k) \preceq \tau^n(l)$,
2. $l \nVdash \psi^n(k) \quad \Leftrightarrow \quad \tau^n(l) \preceq \tau^n(k)$.

Proof. Assume $k \in K$ and $l \in L$ and let, in K^τ, $k' = \tau^n(k)$ and $l' = \tau^n(l)$. Use lemma 2.5.0.5 to conclude that $Th^n(k) = Th^n(k')$ and $Th^n(l) = Th^n(l')$.
1: Observe that $l \Vdash \phi^n(k) \quad \Leftrightarrow \quad l' \Vdash \phi^n(k')$ and hence $l \Vdash \phi^n(k) \quad \Leftrightarrow \quad k' \leq l'$.
2: Likewise, from $l \nVdash \psi^n(k) \quad \Leftrightarrow \quad l' \nVdash \psi^n(k')$ conclude $l \nVdash \psi^n(k) \quad \Leftrightarrow \quad l' \leq k'$. ⊣

2.5.0.12. COROLLARY. *If k is a node in a finite n-model and ψ is an* **IpL***-formula, then:*

$$k \Vdash \psi \quad \Leftrightarrow \quad \phi^n(k) \vdash \psi.$$

Let $K^\phi = \langle \{\phi^n(k) \mid k \in K\}, \vdash \rangle$, then it is easily verified that K^ϕ, with the obvious valuation $\phi^n(k) \Vdash p \quad \Leftrightarrow \quad \phi^n(k) \vdash p$, is a Kripke model. Recall that by definition 2.5.0.6, K^τ is the maximal reduction of K. Now we are ready to state another important (and easy to prove) corollary from theorem 2.5.0.9.

2.5.0.13. COROLLARY. *The model K^τ is isomorphic to the model K^ϕ.*

Readers familiar with [Jankov 68] may wonder why Jankov's name has been connected to theorem 2.5.0.9. The following corollary about finite *frames* presents what is usualy known as Jankov's theorem.

Let us call a reflexive, transitive and anti-symmetric frame an **IpL** frame for short. To define bisimulation between frames, we simply use definition 2.2.0.1 leaving out the condition on the atoms forced. Likewise we may define $k \rightleftarrows l$ between nodes k and l in frames in the obvious way.

2.5.0.14. COROLLARY. *For every finite rooted* **IpL** *frame* $\uparrow k$ *there is a formula* ψ_k *such that for any finite* **IpL** *frame* F *we have:* $F \nVdash \psi_k$ *iff for some* $l \in F$ *it is true that* $k \leftrightarrows l$.

Proof. Define a valuation on $\uparrow k$ on the set of atoms $\{p_i \mid 1 \leq i \leq |\uparrow k|\}$, in such a way that there is a 1–1 mapping $\sigma : \uparrow k \mapsto \{p_i \mid 1 \leq i \leq |\uparrow k|\}$ and for $l \in \uparrow k$ we have $l \Vdash \sigma(m) \quad \Leftrightarrow \quad l \leq m$. For the formula ψ_k in the corollary take $\psi^n(k)$ (assuming $|\uparrow k| = n$). If $F \nVdash \psi^n(k)$ then for some model K based on F we will have for some $l' \in F$ that $l' \nVdash \psi^n(k)$. Now apply the theorem to infer that for some $l \leq l'$ we will have (in K) that $\tau^n(k) = \tau^n(l)$ and hence, by theorem 2.5.0.3, also $k \leftrightarrows l$. $\dashv$

The Jankov theorem was independently proved by De Jongh in his dissertation [De Jongh 68]. In modal logic Fine in [Fine 85] introduced *subframe formulas* for finite transitive frames for which he proved the modal analogue of the Jankov theorem, apparently without being aware of theorems in intuitionistic propositional logic proved by Jankov and De Jongh.

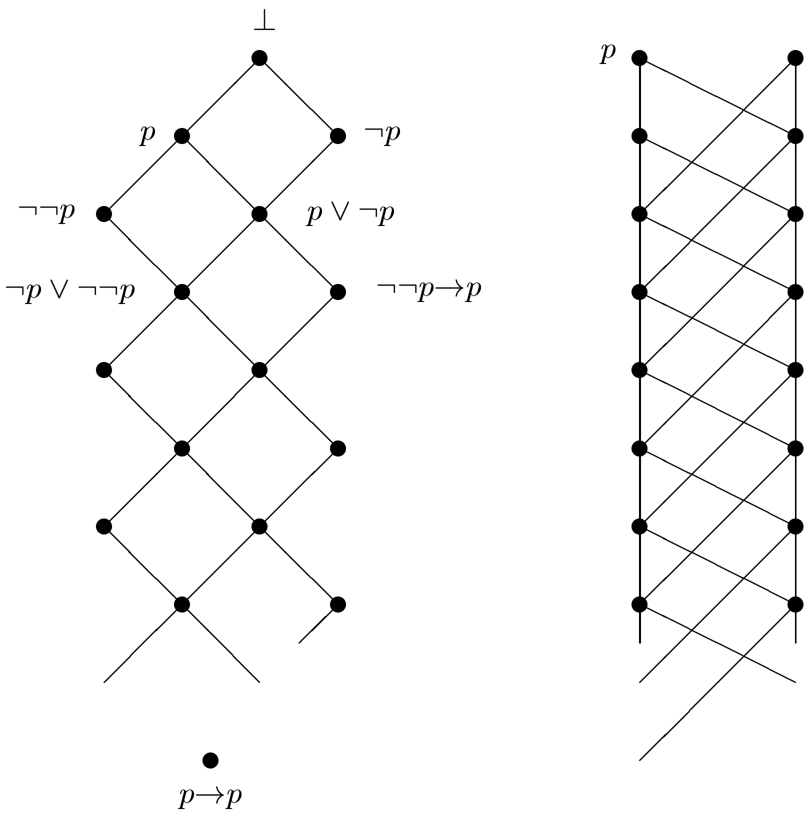
Obviously there are infinitely many types (and semantic types) in fragments $\mathbf{IpL}^n$ if $n \geq 1$.

The diagram of the fragment $\mathbf{IpL}^1$, see figure 6, is known as the Rieger-Nishimura lattice (see [Nishimura 60]) and the ordered set of all non-derivable irreducible types is the exact Kripke model of this fragment (the set of all elements will correspond to $\top$).

Note that all semantic types in $\mathbf{IpL}^1$ are realized in this model. Hence it will be complete for $\mathbf{IpL}^1$. As every semantic type corresponds to an irreducible formula (its type) every finite closed set of irreducible formulas corresponds to an $\mathbf{IpL}^1$ formula (the disjunction of the set of irreducibles). As the set of all elements is the only infinite closed subset of the model and is assigned to $\top$, this proves the model to be the exact model of $\mathbf{IpL}^1$.

For $n > 1$ the fragment $\mathbf{IpL}^n$ will not have an exact model as the diagram of $\mathbf{IpL}^n$ is not a complete distributive lattice for $n > 1$. For example, let $\{\phi_n(p) \mid n \in I\!N\}$ be a set of representatives of the irreducible equivalence classes in $\mathbf{IpL}^1$ and q an atomic formula. Then $\{q \wedge \phi^n(p) \mid \ \nvdash \phi_n(p)\}$ is a closed set of irreducible formulas, that does not correspond to a formula in $\mathbf{IpL}^2$.

2.5.0.15. FACT. *For* $n > 1$ *the fragment* IpL^n *does not have an exact model.*



6. FIGURE. *The diagram*[9] *of* $\mathbf{IpL}^1$ *(left) and its exact Kripke model (right).*

2.6 Calculations in exact models

As explained in the introduction of this chapter and illustrated by the examples of exact models in the previous sections, the proof of the construction of an exact Kripke model K for some fragment F is accompanied by a mapping $[\![.]\!]$ of formulas in F to (closed) subsets of K. A finite exact Kripke model K and its mapping $[\![.]\!]$ together provide us with a decision method for formulas in F. The restriction to closed subsets is necessary only in the case of fragments of **IpL**. In dealing with classical propositional logics (**CpL** or modal systems extending **K**) all subsets of K will be considered to be closed. So in topological terms, in classical logic we use the discrete topology and in intuitionistic logic the topology of upwardly closed subsets induced by the order of K. Recall the definition of the interior operation (rephrased in the context of Kripke models):

2.6.0.1. DEFINITION. *Let K be a Kripke model and $X \subseteq K$. Then X°, the* interior *of X is defined as:*

$$X^\circ = \bigcup\{Y \subseteq X \mid Y \textit{ is closed}\}.$$

[9]Or its dual, according to our definition of $Diag(F)$ in Chapter 1.

In addition the following definition turns out to be useful.

2.6.0.2. DEFINITION. *Let K be a Kripke model and $X \subseteq K$. Then $X^\bullet$, the* predecessor set *of X is defined as:*

$$X^\bullet = \{k \in K \mid \exists l \in X(kRl)\}.$$

It is easily verified that, writing $\overline{X}$ for the complement of set X, in **IpL** models the interior can be calculated as:

$$X^\circ = X \setminus \overline{X}^\bullet.$$

2.6.0.3. FACTS. *Let K be a finite Kripke model for fragment F and let $[\![\phi]\!] = \{k \in K \mid k \Vdash \phi\}$. For all formulas ϕ and ψ of F (and as far as the connectives are applicable in F):*

1. $[\![\phi \wedge \psi]\!] = [\![\phi]\!] \cap [\![\psi]\!]$;
2. $[\![\phi \vee \psi]\!] = [\![\phi]\!] \cup [\![\psi]\!]$;
3. $[\![\phi \to \psi]\!] = ((K \setminus [\![\phi]\!]) \cup [\![\psi]\!])^\circ$;
4. $[\![\neg\phi]\!] = (K \setminus [\![\phi]\!])^\circ$;
5. $[\![\Diamond\phi]\!] = [\![\phi]\!]^\bullet$;
6. $\vdash \phi \quad \Rightarrow \quad [\![\phi]\!] = K$;
7. $\phi \vdash \psi \quad \Rightarrow \quad [\![\phi]\!] \subseteq [\![\psi]\!]$.

All of these facts can be proved by writing out the definitions and using well-known facts about Kripke semantics.

In case K is an exact Kripke model for the fragment F, the implications in the last two facts above can be changed into equivalences.

Hence $[\![\phi]\!]$ can be calculated using set theoretic and topological operations on the exact model.

A computer program to calculate $[\![\phi]\!]$ on an exact Kripke model will need the relevant information about the exact model to calculate the set operations and the predecessor sets. Clearly this can be done in linear time, which makes testing of formulas using exact models such an efficient decision procedure.

For exact models of the fragments of $\mathbf{CpL}^n$ we do not need predecessor sets and the calculation of $[\![\phi]\!]$ is very much like constructing a truth table for ϕ.

The testing of formulas by calculations in an exact Kripke model of a fragment F can be used to calculate the diagram of F and all its subfragments. Let G be a subfragment of F and let K_F be a finite exact model of F. To calculate the diagram of G the algorithm *mkDiag* is given the $[\![p]\!]$ of all atomic formulas in G. These atomic formulas are taken as the representatives of their equivalence classes and the start of a list of elements of the diagram to be constructed. From this list (of formulas representing equivalence classes already found) the algorithm systematically picks one or two representatives to make a new formula according to the connectives available in G. Such a new formula ϕ is taken as a candidate representative of a class not yet in the list. Using the rules explained above $[\![\phi]\!]$ is calculated and compared with the sets corresponding to the classes already found. If ϕ does represent an equivalence class not yet in the list, ϕ is added to the list of representatives and

$[\![\phi]\!]$ to the list of sets corresponding to the representatives. As the diagram of G is finite, this procedure terminates. In fact, by testing for each representative ϕ both $[\![\phi]\!] \subseteq [\![\psi]\!]$ and $[\![\psi]\!] \subseteq [\![\phi]\!]$ the algorithm does not only determine whether ϕ represents a new class or not, but also keeps track of the relations in the diagram.

2.7 Games and bisimulations

Let us finish this chapter with the introduction of *Ehrenfeucht* games and their relation to (layered) bisimulations and to semantic types in general. In the next chapter we will occasionally use this kind of game to decide the equivalence of nodes in Kripke models for formulas in certain fragments of **IpL**. For an application of Ehrenfeucht games to second-order and intensional logic see [Doets 87].

In the present context an *Ehrenfeucht game* is a game with two Kripke models, played by two players (player I and player II). At the start player I makes a choice between the two models, by pointing to a world in one of the models. After this start of the game, each of the players in turn will point to a world in the player's model. If a player has chosen world l as the previous move, the l' for the present move has to fulfill the condition that $l \leq l'$. If player I made a move by choosing world k, the world l in the move of player II will also have to meet the condition that $atom(k) = atom(l)$.

The game is finished is one of the players is unable to come up with a satisfactory world. A player that cannot make a valid move in turn has lost.

The idea behind this kind of game is simple. Player II will win the game if able to simulate each of the moves of player I. As player I may choose models first, a *winning strategy* for player II is only possible if there is a simulation relation between the models.

We will use $G(K, L)$ for the Ehrenfeucht game with models K and L defined by the rules above. For player II having a winning strategy we introduce the notation $\models G(K, L)$.

2.7.0.1. FACT. *Let $G(K, L)$ be an Ehrenfeucht game for Kripke models K and L. Then $\models G(K, L)$ iff there exists a bisimulation S between K and L and S is full ($dom(S) = K$ and $ran(S) = L$).*

This fact is a simple consequence of the similarity between the definition of an Ehrenfeucht game above and the definition of a bisimulation in the preliminaries of this chapter.

In the sequel the Ehrenfeucht games all will be played on finite n-models. Our first (simple) refinement of this general scheme of Ehrenfeucht games will be the introduction of two *starting worlds*.

In a game $G(K, L, \langle k, l\rangle)$ with starting worlds $k \in K$ and $l \in L$ (and K and L finite n-models) the first move of each player has to be either k or l. As an easy corollary of fact 2.7.0.1 we have $\models G(K, L, \langle k, l\rangle)$ iff $k \rightleftarrows^n l$.

Other refinements of the scheme of Ehrenfeucht games will be introduced in Chapter 3.

Chapter 3

Exact Models in IpL

3.1 Introduction

In this chapter we will describe all non-trivial finite fragments of intuitionistic propositional logic with atoms in some finite set $\{p_1, \ldots, p_n\}$ and connectives in the set $\{\wedge, \vee, \rightarrow, \neg, \neg\neg\}$. Each of these fragments will be denoted by the number of atoms and the set of connectives used, like $[\wedge, \vee]^2$ for the fragment with two atoms and conjunction and disjunction as its only connectives. Not included are the descriptions of the trivial fragments $[\neg]^n$, $[\neg\neg]^n$ and the fragment with n atomic formulas and no connectives.

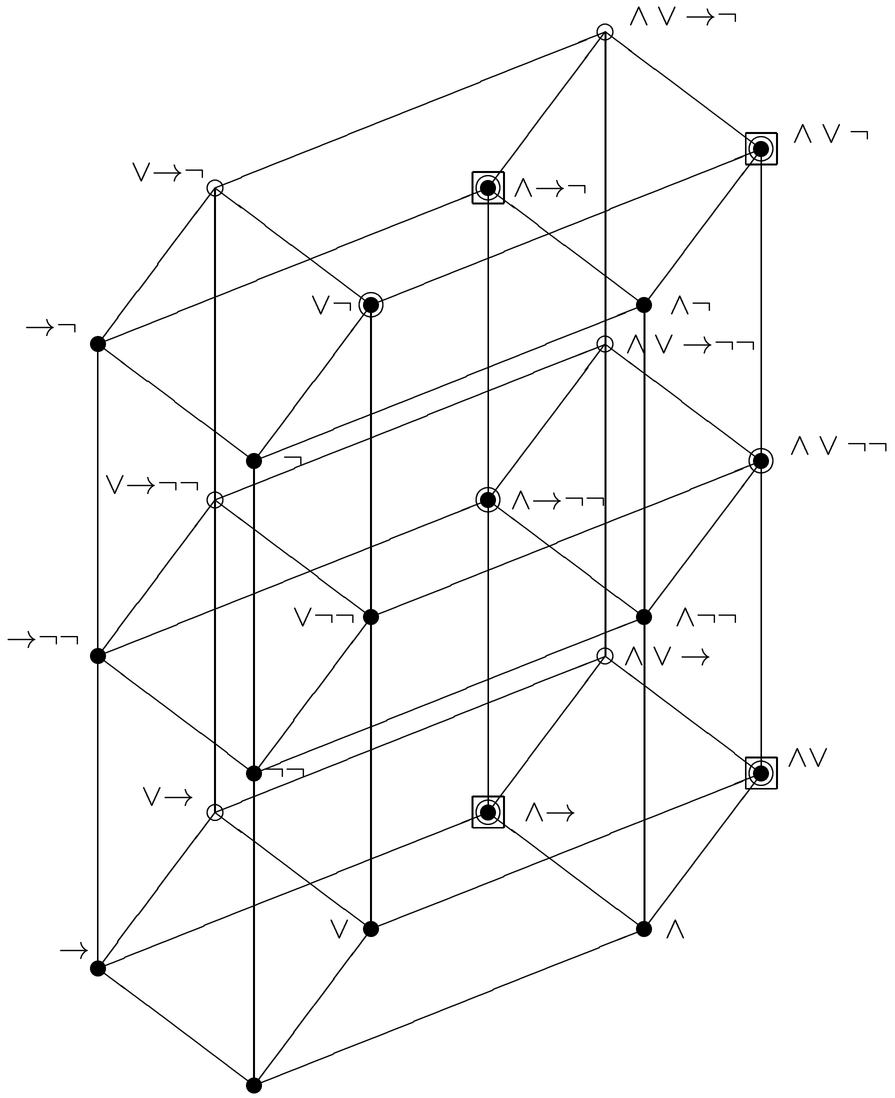
Our main task in this chapter will be to show how to construct exact Kripke models for fragments of $\mathbf{IpL}^n$, using the notion of semantic type introduced in Chapter 2. In some cases (i.c. the fragments $[\vee, \neg]^n$ and fragments with $\neg\neg$ without $\neg$) there exists an exact model, but no exact Kripke model. For these fragments we will show to construct, via a *completion* of the exact model, a *universal model* that can be used to calculate the diagram (and subdiagrams). Such a completion of the exact model will be called a *Kripke completion.*

In the sequel we will define, for each of the fragments F in $\mathbf{IpL}^n$ with an exact model, semantic types $\tau_F(k)$ and corresponding types $\phi_F(k)$ for nodes k in a Kripke model. As it should be clear from the context which is the fragment in question, we will most often drop the index F.

The fragments of **IpL** with connectives in the set $\{\wedge, \vee, \rightarrow, \neg, \neg\neg\}$ can be pictured as a lattice (using the inclusion of fragments defined in the preliminaries of the introduction).

The lattice of fragments in the picture below (which was also given in the general introduction in Chapter 1) provides us with an overview of the (non-trivial) fragments that can be obtained by restricting the set of connectives.

Recall that fragments with an infinite diagram (at least in case of more than one propositional variable) are denoted by an open circle. Finite fragments (in $\mathbf{IpL}^n$) are pictured as closed circles and fragments with an exact model have a closed circle surrounded by an additional open circle. Fragments with an exact Kripke model are marked by a square.



7. FIGURE. *The lattice of fragments in* **IpL**.

As was pointed out in Chapter 2, the diagrams of fragments with an exact Kripke model can be calculated very efficiently if the model is given. The same is true for the subfragments of fragments with an exact Kripke model (just restrict the calculations of formulas and sets according to the restrictions in the subfragment).

As observed in the introduction of Chapter 2, a fragment of **IpL** has a finite exact model iff its diagram is a finite, distributive lattice.

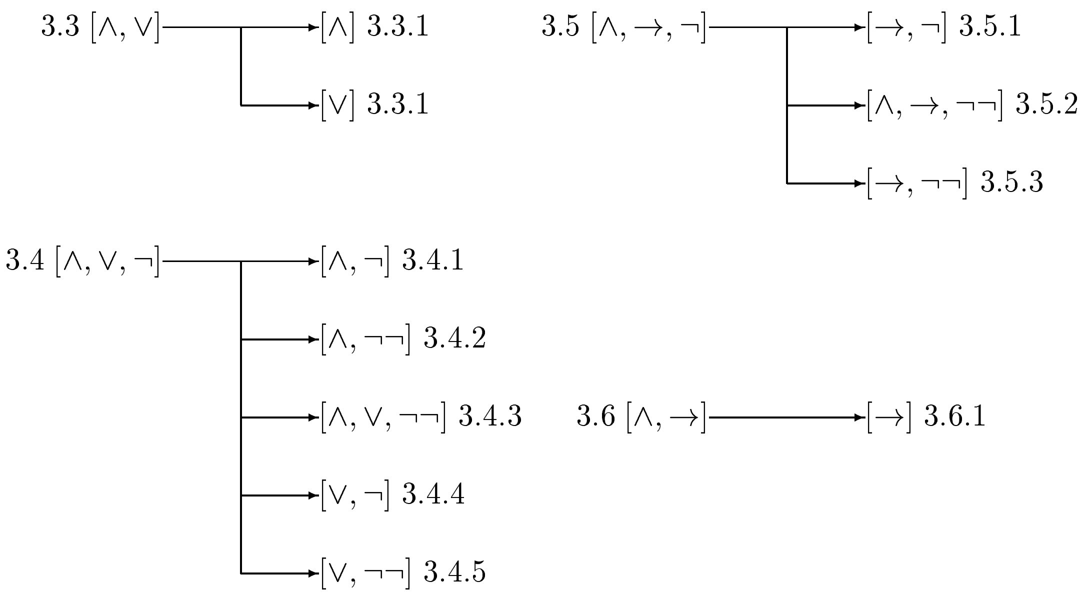
This criterion may be necessary and sufficient for the existence of an exact model for a fragment of $\mathbf{IpL}^n$, but it does not reveal how to obtain an exact model for a particular fragment.

If we knew the irreducible formulas in a fragment F, we could order them with $\dashv$ to obtain $Exm(F)$, the exact model of F. But determining the irreducible formulas in

F may be far from easy. For example in fragments not containing the disjunction (like the $[\wedge, \to, \neg]$ fragments) it is not immediately clear which formulas are irreducible in the lattice of $Diag(F)$.

In the sequel of this chapter we will construct exact models for fragments F by defining an appropriate semantic type and a (straightforward) ordering.

Except for the preliminaries, this chapter has four sections. In the first subsection of each section we describe one of the fragments with an exact Kripke model. The other subsections deal with subfragments that do not have an exact Kripke model of their own.



8. FIGURE. *The structure of this chapter.*

The general structure of a subsection about fragment F is first to define a class of Kripke models $\mathcal{M}$, for which the fragment is complete. We then define semantic types $\tau_F(k)$ and type formulas $\phi_F(k)$ for the nodes k in the Kripke models in $\mathcal{M}$. In general, this set of semantic types in F can be turned into a (minimal) complete model for F. This *universal* model for F will contain an exact model, if such a model exists for F. In case F has an exact Kripke model, the exact Kripke model and the universal model will coincide.

3.2 Preliminaries

If F is a fragment of **IpL** with a finite exact model, the elements in such a model correspond to the irreducible formulas in $Diag(F)$, as observed in Chapter 2. For

IpL fragments F this implies that if an exact model exists, it is unique (up to isomorphism).

To prove this, we will show that the order in the exact model is determined by the derivability relation. Let ϕ and ψ be irreducible formulas in F and let k_ϕ and k_ψ denote the corresponding nodes in an exact model $Exm(F)$. If ω is the correspondence between formulas and closed subsets in $Exm(F)$ then clearly:

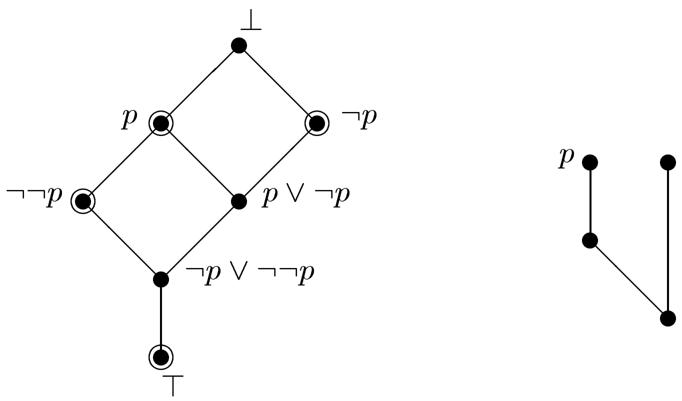
$$\phi \vdash \psi \quad \Leftrightarrow \quad \omega(\phi) \subseteq \omega(\psi) \quad \Leftrightarrow \quad k_\psi \leq k_\phi.$$

A fortiori this is true if F has an exact Kripke model.

3.2.0.1. FACT. *If F is a fragment in* **IpL** *and F has an exact (Kripke) model, then this model is unique up to isomorphism.*

Because of this fact we will in the sequel, when dealing with exact models in **IpL** fragments, simply write 'the' exact (Kripke) model instead of 'an' exact (Kripke) model.

As an example of the relationship between the diagram and the exact (Kripke) model of a fragment, figure 9 shows the diagram and the exact Kripke model of the fragment $[\wedge, \vee, \neg]^1$, where the irreducible elements in the diagram are marked with an extra circle.



9. FIGURE. *The diagram of $[\wedge, \vee, \neg]^1$ (left) and its exact Kripke model (right).*

In case $\vee$ is in the **IpL** fragment F, $\vee$ will naturally act as the join in the diagram of F. Hence the irreducibles in F will be the $\vee$-irreducible formulas (i.e. those formulas ϕ in F such that for all ψ and χ in F, $\phi \vdash \psi \vee \chi$ implies $\phi \vdash \psi$ or $\phi \vdash \chi$).

To characterize the $\vee$-irreducible formulas in **IpL** we will use the *Aczel slash* (see for example [TD 88]).

3.2.0.2. DEFINITION. (Aczel slash) *Let Γ be a set of* **IpL** *formulas. For an* **IpL** *formula ϕ define $\Gamma \mid \phi$ inductively as:*

1. $\Gamma \mid p \quad \Leftrightarrow \quad \Gamma \vdash p$ *for p atomic or $p = \perp$;*
2. $\Gamma \mid \phi \wedge \psi \quad \Leftrightarrow \quad \Gamma \mid \phi$ *and* $\Gamma \mid \psi$*;*
3. $\Gamma \mid \phi \vee \psi \quad \Leftrightarrow \quad \Gamma \mid \phi$ *or* $\Gamma \mid \psi$*;*
4. $\Gamma \mid \phi \rightarrow \psi \quad \Leftrightarrow \quad \Gamma \vdash \phi \rightarrow \psi$ *and* $(\Gamma \mid \phi \quad \Rightarrow \quad \Gamma \mid \psi)$.

3.2.0.3. FACTS. *Let Γ be a set of* **IpL** *formulas and let ϕ and ψ be* **IpL** *formulas.*

1. *([Kleene 62]) If $\phi \not\equiv \bot$ then ϕ is $\vee$-irreducible iff $\phi \mid \phi$.*
2. *If $\Gamma \mid \phi$ then $\Gamma \vdash \phi$.*
3. *If $\Gamma \vdash \phi{\to}\psi$ and $\Gamma \nvdash \phi$ then $\Gamma \mid \phi{\to}\psi$.*
4. *All formulas in $[\wedge,\to,\neg]$ are either equivalent to $\bot$ or $\vee$-irreducible.*
5. *All formulas $\neg\phi$ not equivalent to $\bot$ are $\vee$-irreducible.*

Especially the last two of the above facts will be useful in this chapter.

In the rest of this chapter the Kripke models used will be **IpL** models (reflexive, transitive and anti-symmetric). In particular, if we mention n-models in this section we mean **IpL** n-models.

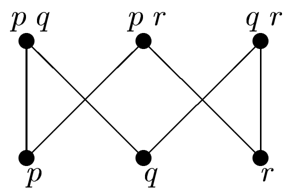
3.3 The $[\wedge,\vee]$ fragments

The structure of the $[\wedge,\vee]^n$ fragments is relatively well known (see [DP 90] for example):

3.3.0.1. FACTS. *Let $\vdash_c$ be the derivability relation in* **CpL***.*

1. *The $[\wedge,\vee]^n$ fragments in* **IpL** *and* **CpL** *coincide. For formulas ϕ and ψ in $[\wedge,\vee]$:*
$$\phi \vdash \psi \quad \Leftrightarrow \quad \phi \vdash_c \psi.$$
2. *The diagram of $[\wedge,\vee]^n$ is isomorphic to the free distributive lattice over n generators.*
3. *Each $\phi \in [\wedge,\vee]^n$ is equivalent to a finite disjunction of $[\wedge]^n$ formulas.*
4. *All $[\wedge]^n$ formulas are $\vee$-irreducible.*
5. *The diagram of $[\wedge]^n$ is dual to the diagram of $[\vee]^n$.*
6. *For all $\phi \in [\wedge,\vee]^n$ we have $\bigwedge\{p_1,\ldots,p_n\} \vdash \phi$.*

From 3, it follows that the diagram of $[\wedge]^n$ is almost the exact model of $[\wedge,\vee]^n$. Almost, as the empty set does not correspond to a formula in $[\wedge,\vee]^n$. On the other hand, the conjunction of all atoms in $[\wedge,\vee]^n$ is the bottom element of the diagram. By removing this bottom element from the diagram of $[\wedge]^n$ we get the exact model of $[\wedge,\vee]^n$ (where the empty subset of the exact model corresponds to the bottom of the diagram).



10. FIGURE. *The exact Kripke model of $[\wedge,\vee]^3$.*

The model above has 18 closed subsets, from which we may infer that the diagram of $[\wedge,\vee]^3$ has 18 elements.

Note that for a node k in a Kripke model K the formula $\bigwedge atom^n(k)$ will be the $[\wedge,\vee]^n$-type of k, an axiom of $Th^n(k)$, the $[\wedge,\vee]^n$ theory of k.

3.3.0.2. DEFINITION. *Let k be a node in a Kripke model. The semantic type of k in $[\wedge,\vee]^n$, $\tau^n(k)$ is defined as:*

$$\tau^n(k) = \langle atom^n(k), \emptyset\rangle.$$

If t and t' are semantic types in $[\wedge,\vee]^n$, define:

$$t \preceq t' \quad\Leftrightarrow\quad j_0(t) \subseteq j_0(t').$$

The type formula of k in $[\wedge,\vee]^n$, $\phi^n(k)$ is defined as:

$$\phi^n(k) = \bigwedge j_0(\tau^n(k)).$$

The following lemma states that the above defined types are indeed semantic types in $[\wedge,\vee]^n$ as described in Chapter 2.

3.3.0.3. LEMMA. *If k and l are nodes in Kripke models, then*

$$l \Vdash \phi^n(k) \quad\Leftrightarrow\quad \tau(k) \preceq \tau(l) \quad\Leftrightarrow\quad Th^n(k) \subseteq Th^n(l) \quad\Leftrightarrow\quad \phi^n(l) \vdash \phi^n(k).$$

Proof. Obvious. ⊣

It is also obvious that if k and l are nodes in an **IpL** Kripke model K, then $k \leq l$ implies $\tau(k) \preceq \tau(l)$.

Note that the type $\langle\{p_1,\ldots,p_n\},\emptyset\rangle$ is a special one in $[\wedge,\vee]^n$ in that a node with such a type will force all formulas in the fragment. We will encounter such *bottom types* again in the sequel and they will be disregarded in the construction of the exact Kripke model (or the universal model in some cases). The reason has been stated above already, for including such a type would prevent the empty set in the exact model to correspond to the bottom of the diagram.

3.3.0.4. THEOREM. *The set of types in $[\wedge,\vee]^n$, with exception of the bottom type, i.e. $\langle\{p_1,\ldots,p_n\},\emptyset\rangle$, ordered by $\preceq$ and taking $atom^n(t) = j_0(t)$ for a type t, is the exact Kripke model of $[\wedge,\vee]^n$.*

Proof. From the lemma above and the observations following it, it should be clear that in the intended model each t realizes its own type, (i.e. $\tau^n(t) = t$). Moreover, we have $t \preceq t'$ iff for all formulas in $[\wedge,\vee]^n$ it is true that $t \Vdash \phi \;\Rightarrow\; t' \Vdash \phi$. Hence $[\![\phi]\!] = \{t \mid t \Vdash \phi\}$ is a $1-1$ correspondence between closed subsets of the model and formulas in $[\wedge,\vee]^n$. ⊣

In general the exact Kripke model of $[\wedge,\vee]^n$ will have $2^n - 2$ nodes (as there are $2^n - 2$ nonempty proper subsets of a set of n elements).

Obviously the types in $[\wedge,\vee]$ are just sets of atoms if we disregard the general format of semantic types. Hence the exact Kripke model above is isomorphic to the set of proper nonempty subsets of $\{p_1,\ldots,p_n\}$, ordered by inclusion.

As the characteristic functions of closed sets in the exact Kripke model of $[\wedge,\vee]^n$ are the monotonic functions into $\{0,1\}$, theorem 3.3.0.4 establishes the correspondence between formulas of $[\wedge,\vee]^n$ and monotonic functions of $2^n \mapsto 2$. The problem of determining the number $D(n)$ of these functions (for each n) goes back to Dedekind and is known in a different, but equivalent, form as the Sperner problem (see [Kleitman 69], [Kisielewicz 88]).

In [Sloane 73] there is a table[1] (nr. 1 439) for $D(n)$:

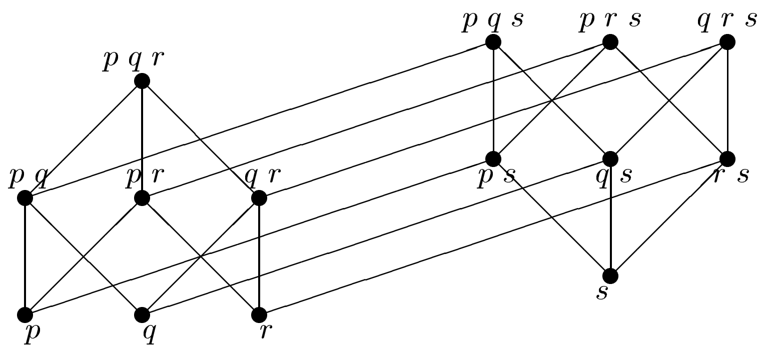
n	$D(n)$
1	1
2	4
3	18
4	166
5	7 579
6	7 828 352
7	2 414 682 040 996

Although there is no simple formula known to calculate the number $D(n)$ there is a simple construction for the exact model of $[\wedge,\vee]^{n+1}$ from the exact model of $[\wedge,\vee]^n$.

Let E^n be the exact model of $[\wedge,\vee]^n$. To obtain the exact model E^{n+1}, take a copy of E^n, denoted as E^n_{n+1}, and connect every $k\in E^n$ with its twin in $k'\in E^n_{n+1}$ (hence $k<k'$). Now change the valuation on E^n_{n+1}, so that in every node $l\in E^n_{n+1}$ also the atom p_{n+1} is forced. Next add a new root below E^n_{n+1} where only p_{n+1} is forced and add a new node k above all nodes in E^n in such a way that $atom(k)=\{p_1,\ldots,p_n\}$.

Clearly the new model exactly realizes all types in $[\wedge,\vee]^{n+1}$, but for the bottom type (where all atoms in the fragment would be forced).

The procedure is illustrated in the figure below, where the exact Kripke model of $[\wedge,\vee]^4$ is constructed from the exact Kripke model of $[\wedge,\vee]^3$.



11. FIGURE. *The exact Kripke model of* $[\wedge,\vee]^4$.

[1] It is convenient to define $D(0)=0$.

Note that from this construction it simply follows that the exact Kripke model of $[\wedge, \vee]^n$ is the *n-dimensional hypercube* without its top and bottom elements.

3.3.1 $[\wedge]$ and $[\vee]$ fragments

The diagram of $[\wedge]^n$, and dually $[\vee]^n$, is of course isomorphic to the powerset of the nonempty subsets of a set of n elements, ordered by inclusion. Hence the diagram of $[\wedge]^n$ (or $[\vee]^n$) will be isomorphic to the n-dimensional hypercube without its bottom element and have $2^n - 1$ elements.

3.4 The $[\wedge, \vee, \neg]$ fragments

Let us start the treatment of the $[\wedge, \vee, \neg]$ fragments by defining an Ehrenfeucht game for this fragment (see definition 2.7).

3.4.0.1. Definition. *Let K and L be finite Kripke models, $k \in K$ and $l \in L$. The* Ehrenfeucht game *for $[\wedge, \vee, \neg]^n$ with starting worlds k and l, $G^n(K, L, \langle k, l\rangle)$, is a game between two players, I and II, who each make exactly one move, in turn.*

Player I starts by choosing a terminal node m_I above either k or l. Player II replies by choosing a terminal node m_{II} above k, if $l \leq_L m_I$, or above l, if $k \leq_K m_I$.

Player II has won the game if $atom^n(k) = atom^n(l)$ and $atom^n(m_I) = atom^n(m_{II})$.

$\models G^n(K, L, \langle k, l\rangle)$ will denote that there is a winning strategy for player II in the game $G^n(K, L, \langle k, l\rangle)$.

Let $Th^n(k)$ denote the $[\wedge, \vee, \neg]^n$ theory of node k. For finite Kripke models K and L (and $k \in K$, $l \in L$), we have the following theorem.

3.4.0.2. Theorem. $\models G^n(K, L, \langle k, l\rangle) \quad \Leftrightarrow \quad Th^n(k) = Th^n(l)$

Proof. $\Rightarrow$: By induction on the length of $\phi \in [\wedge, \vee, \neg]^n$ we will prove that $k \Vdash \phi \quad \Leftrightarrow \quad l \Vdash \phi$. The cases where ϕ is either atomic, a conjunction or a disjunction are trivial. Assume $\phi = \neg\psi$ and $k \Vdash \phi$. Then for no terminal node m above k it will be true that $m \Vdash \psi$. Suppose m_I is a terminal node such that $l \leq m_I$. If $m_I \Vdash \psi$ then, as II has a winning strategy for the game $G(K, L, \langle k, l\rangle)$, there is a terminal node $m_{II} \geq k$ such that $atom^n(m_I) = atom^n(m_{II})$. Which would imply $m_{II} \Vdash \psi$, a contradiction. This proves that for no terminal node $m_I \geq l$ $m_I \Vdash \psi$, and hence $l \Vdash \neg\psi$.

$\Leftarrow$: Note that $Th^n(k) = Th^n(l)$ implies $atom^n(k) = atom^n(l)$. Suppose player I chooses m_I (say in K, above k). Recall the definition of $\phi^n_{\mathbf{CpL}}(m_I)$ (definition 2.3.0.2). Then $k \nVdash \neg\phi^n_{\mathbf{CpL}}(m_I)$ and, as $\neg\phi_{\mathbf{CpL}}(m_I)$ is equivalent to a formula in $[\wedge, \vee, \neg]^n$ and $Th^n(k) = Th^n(l)$, $l \nVdash \neg\phi^n_{\mathbf{CpL}}(m_I)$. So, for some terminal node $m_{II} \geq l$, $m_{II} \Vdash \phi^n_{\mathbf{CpL}}(m_I)$, which implies $atom^n(m_I) = atom^n(m_{II})$. Hence there is a winning strategy for II in the game $G(K, L, \langle k, l\rangle)$. $\dashv$

The proof of the theorem above contains both a suggestion for the definition of $\tau^n(k)$, the semantic type in $[\wedge,\vee,\neg]^n$, and of $\phi^n(k)$, the type in $[\wedge,\vee,\neg]^n$ (i.e. an axiom for $Th^n(k)$). Recall the definition of $\phi^n_{\mathbf{CpL}}(k)$ from definition 2.3.0.1.

3.4.0.3. DEFINITION. *Let k be a node in a finite Kripke model and let $Ter(k)$ denote the set of terminal nodes above k:*

$$Ter(k) = \{m \geq k \mid m \text{ is a terminal node}\}.$$

Define:

$$\tau^n(k) = \begin{cases} \langle atom^n(k), \emptyset\rangle & \text{if } \quad \forall l > k.\, atom^n(l) = atom^n(k) \\ \\ \langle atom^n(k), \{\tau^n(l) \mid l \in Ter(k)\}\rangle & \text{otherwise.} \end{cases}$$

For semantic types t and t' in $[\wedge,\vee,\neg]^n$ define:

$$t \preceq t' \quad \Leftrightarrow \quad t = t' \text{ or } t' \in j_1(t) \text{ or } (j_0(t) \subseteq j_0(t') \text{ and } \emptyset \neq j_1(t') \subseteq j_1(t))$$

$$\phi^n(k) = \begin{cases} \phi^n_{\mathbf{CpL}}(k) & \text{if } \forall l > k.\, atom^n(l) = atom^n(k) \\ \\ \bigwedge j_0(\tau^n(k)) \wedge & \\ \neg\neg \bigvee \{\phi^n_{\mathbf{CpL}}(l) \mid \tau^n(l) \in j_1(\tau^n(k))\} & \text{otherwise.} \end{cases}$$

Observe that in particular $\tau^n(k) = \langle atom^n(k), \emptyset\rangle$ if k is a terminal node.

The next lemma shows we are on the right track with these characterizations of the $[\wedge,\vee,\neg]^n$ theory of a node in a Kripke model.

But let us first state as a fact the following simple consequence of the definition of a semantic $[\wedge,\vee,\neg]^n$ type.

3.4.0.4. FACT. *If $k \in K$ and $l \in L$ are nodes in finite Kripke models and $\tau^n(k) = \tau^n(l)$, then $\models G(K, L, \langle k, l\rangle)$.*

The next lemma, in combination with theorem 3.4.0.2, has as a consequence that for nodes $k \in K$ and $l \in L$ in finite Kripke models K and L also $\models G(K, L, \langle k, l\rangle)$ implies $\tau^n(k) = \tau^n(l)$.

3.4.0.5. LEMMA. *Let k and l be nodes in finite Kripke models. Then the following statements are equivalent:*

1. $l \Vdash \phi^n(k)$;
2. $\tau^n(k) \preceq \tau^n(l)$;
3. $Th^n(k) \subseteq Th^n(l)$;
4. $\phi^n(l) \vdash \phi^n(k)$.

Proof. We will prove $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 1$.

$1 \Rightarrow 2$: Assume $l \Vdash \phi^n(k)$. If $\phi^n(k) = \phi^n_{\mathbf{CpL}}(k)$, then clearly for all $m \geq l$ we have $atom^n(m) = atom^n(k)$ and hence $\tau^n(l) = \langle atom^n(k), \emptyset\rangle = \tau^n(k)$. On the other hand, if $\tau^n(k) = \langle atom^n(k), \{\tau^n(l) \mid l \in Ter(k)\}\rangle$, then for $l' \in Ter(l)$ we can prove that $\tau^n(l') \in j_1(\tau^n(k))$. Let $l' \in Ter(l)$. Then we have, by the definition of $\phi^n(k)$, that $l' \Vdash \bigvee\{\phi^n_{\mathbf{CpL}}(m) \mid \tau^n(m) \in j_1(\tau^n(k))\}$. Hence $l' \Vdash \phi^n_{\mathbf{CpL}}(m)$ for some $m \in Ter(k)$ and, as above, infer that $\tau^n(l') = \tau^n(m)$. Note that either $\tau^n(l) = \tau^n(l')$ for some $l' \in Ter(l)$ and thus $\tau^n(l') \in j_1(\tau^n(k))$, or $j_1(\tau^n(l)) \neq \emptyset$ and we proved $j_1(\tau^n(l)) \subseteq j_1(\tau^n(k))$. In both cases we may conclude $\tau^n(k) \preceq \tau^n(l)$ as trivially $atom^n(k) \subseteq atom^n(l)$ holds if $l \Vdash \phi^n(k)$.

$2 \Rightarrow 3$: Assume $\tau^n(k) \preceq \tau^n(l)$. Note that if for all $m > k$ we have $atom^n(m) = atom^n(k)$, then from $\tau^n(k) \preceq \tau^n(l)$ we may infer that $\tau^n(k) = \tau^n(l) = \langle atom^n(k), \emptyset\rangle$ and hence by fact 3.4.0.4 $\models G(K, L', \langle k, l\rangle)$. Which proves $Th^n(k) = Th^n(l)$, using theorem 3.4.0.2.

So assume there is a $k' \in Ter(k)$ with $atom^n(k') \neq atom^n(k)$. Let $l \in L$ and let L' be the model constructed from L by adding a new node l_0 with $atom^n(l_0) = atom^n(k)$ and placed below l and all terminal nodes above k. Note that such a construction of L' as a finite Kripke model is possible as $atom^n(k) \subseteq atom^n(l)$, which we may infer from the assumption. Also from the assumption that $\tau^n(k) \preceq \tau^n(l)$ we may conclude that $\tau^n(l_0) = \tau^n(k)$. By fact 3.4.0.4 this implies $\models G(K, L', \langle k, l_0\rangle)$ and hence by theorem 3.4.0.2, $Th^n(k) = Th^n(l_0)$. From the construction of L' infer that as a consequence we have $Th^n(k) \subseteq Th^n(l)$.

$3 \Rightarrow 4$: Note that from the two previous steps we may conclude that $\phi^n(m)$ is an axiom of $Th^n(m)$ (for any node m in a finite Kripke model). For suppose $\phi \in Th^n(k)$ and $l \Vdash \phi^n(k)$. Then by combining the first and the second part of this proof we have $Th^n(k) \subseteq Th^n(l)$ and hence $l \Vdash \phi$. Which, by the completeness theorem, proves $\phi^n(k) \vdash \phi$.
From the fact that $\phi^n(m)$ is an axiom for $Th^n(m)$ one easily proves that the inclusion of the theories $Th^n(k) \subseteq Th^n(l)$ implies the interderivability of their axioms: $\phi^n(l) \vdash \phi^n(k)$.

$4 \Rightarrow 1$: As $\phi^n(l)$ is the axiom of $Th^n(l)$, we know that $l \Vdash \phi^n(l)$. And hence from $\phi^n(l) \vdash \phi^n(k)$ we infer $l \Vdash \phi^n(k)$. $\dashv$

From the definition of semantic types in $[\wedge, \vee, \neg]^n$ it is clear that there are only finitely many of these types. It is also easy to prove that all tuples of the form $\langle S, T\rangle$ such that:

1. T is a set of types $\langle U, \emptyset\rangle$, where $U \subseteq \{p_1, \ldots, p_n\}$,
2. if $T \neq \emptyset$ then $S \subseteq \bigcap\{j_0(t) \mid t \in T\}$,
3. if $T = \{\langle U, \emptyset\rangle\}$ then $S \neq U$,

are types in $[\wedge, \vee, \neg]^n$.

As each semantic type of $[\wedge, \vee, \neg]$ can be realized in a rooted **IpL** model with depth less than two, we have established the following fact.

3.4.0.6. FACT. *$[\wedge, \vee, \neg]$ is complete for rooted* **IpL** *models of depth less than two.*

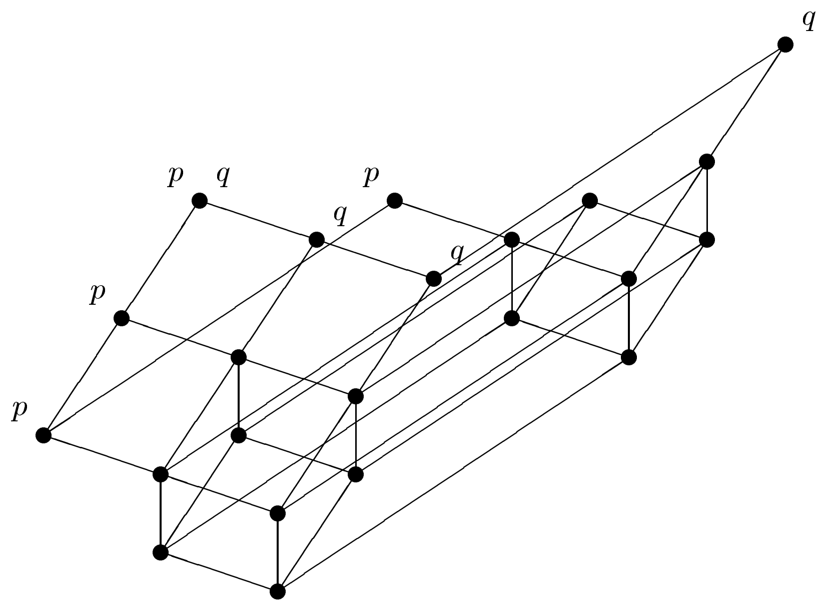
An intermediate logic is a *conservative extension* of a fragment of **IpL** if for any two formulas ϕ and ψ in the fragment ψ is a consequence of ϕ in the intermediate logic iff $\phi \vdash_{\mathbf{IpL}} \psi$. The intermediate logic **IpL** $+$ $((p \to (((q \to r) \to q) \to q)) \to p) \to p$, complete for models of depth less than two, can be proved to be a maximal conservative extension of $[\wedge, \vee, \neg]$. But it is not unique. A. Chagrov announced a proof for the existence of continuum of maximal conservative extensions of $[\wedge, \vee, \neg]^n$ for each $n > 1$.

Ordering the types in $[\wedge, \vee, \neg]^n$, putting $\langle S, T\rangle \preceq \langle S', T'\rangle$ if $S \subseteq S'$ and $T' \subseteq T$ will yield a Kripke model $Exm([\wedge, \vee, \neg]^n)$ (with $atom^n(t) = j_0(t)$).

Note that as $Exm([\wedge, \vee, \neg]^n)$ realizes all semantic types in $Exm([\wedge, \vee, \neg]^n)$, it is a complete Kripke model for this fragment. As a consequence of the above lemma also $[\![\phi^n(k)]\!] = \uparrow k$. Hence every closed subset of $Exm([\wedge, \vee, \neg]^n)$ can be obtained as the valuation of a formula in $[\wedge, \vee, \neg]^n$. Which proves the following theorem.

3.4.0.7. THEOREM. *The model $Exm([\wedge, \vee, \neg]^n)$ defined above is the exact Kripke model of $[\wedge, \vee, \neg]^n$.*

As an example, in figure 12 we give the exact Kripke model of $[\wedge, \vee, \neg]^2$.



12. FIGURE. *The exact Kripke model of $[\wedge, \vee, \neg]^2$.*

As all types in $[\wedge, \vee, \neg]^n$ are formulas in $[\wedge, \neg]^n$, we have the following corollary.

3.4.0.8. COROLLARY. (The $[\wedge, \neg]$ normal form) *In $[\wedge, \vee, \neg]^n$ each formula is equivalent to a disjunction of formulas in $[\wedge, \neg]^n$.*

Note that as each formula in $[\wedge, \neg]^n$ which is not equivalent to $\perp$ is irreducible (use fact 3.2.0.3.4), each of these formulas will be (equivalent to) a type in $[\wedge, \vee, \neg]^n$. As a result, we may state the following fact.

3.4.0.9. FACT. *By leaving out $\perp$, the diagram of $[\wedge, \neg]^n$ becomes the exact Kripke model of $[\wedge, \vee, \neg]^n$.*

We will use the exact Kripke model of $[\wedge,\vee,\neg]^n$ in the proof of the characterization of the $[\wedge,\vee,\neg]$ formulas in **IpL**. First we introduce the *terminal reduction* of a rooted Kripke model.

3.4.0.10. DEFINITION. *For a finite Kripke model K with root k, the submodel $(\uparrow k)^T$, with domain $\{k\}\cup Ter(k)$ and the accessibility relation and valuation inherited from K is called the* terminal reduction *of K.*

Obviously, for a node k in a finite Kripke model K, the semantic type (in $[\wedge,\vee,\neg]^n$) of k in K and in $(\uparrow k)^T$, the terminal reduction of the submodel $\uparrow k$, are the same.

Hence, a rooted Kripke model and its terminal reduction force the same $[\wedge,\vee,\neg]$ formulas (have the same $[\wedge,\vee,\neg]$ theory).

3.4.0.11. THEOREM. *An* **IpL** *formula ϕ is equivalent to a $[\wedge,\vee,\neg]$ formula iff for every node k in a finite Kripke model:*

$$k \Vdash \phi \quad \Leftrightarrow \quad (\uparrow k)^T \Vdash \phi.$$

Proof. As observed above, the node k in a finite Kripke model K and the root of the terminal reduction $(\uparrow k)^T$, have the same semantic type in $[\wedge,\vee,\neg]^n$. By lemma 3.4.0.5 this implies that k and $(\uparrow k)^T$ force the same $[\wedge,\vee,\neg]$ formulas. Which proves one direction of the theorem.

For the other direction, assume ϕ is a formula in $\mathbf{IpL}^n$ and for every k in a finite Kripke model it is true that $k \Vdash \phi \;\Leftrightarrow\; (\uparrow k)^T \Vdash \phi$. Let χ be the $[\wedge,\vee,\neg]$ formula with $[\![\chi]\!] = [\![\phi]\!]$ in $Exm([\wedge,\vee,\neg]^n)$.

We will show that ϕ is equivalent to χ by showing (for k a node in a finite Kripke model) $k \Vdash \phi \;\Leftrightarrow\; k \Vdash \chi$. We first use the assumption that $k \Vdash \phi$ is equivalent to $(\uparrow k)^T \Vdash \phi$. The root k in $(\uparrow k)^T$ clearly bisimulates the node $\tau^n(k)$ in the terminal reduction $(\uparrow \tau^n(k))^T$ of the submodel $\uparrow \tau^n(k)$ in the exact Kripke model $Exm([\wedge,\vee,\neg]^n$. Hence, $(\uparrow k)^T \Vdash \phi$ is equivalent to $(\uparrow \tau^n(k))^T \Vdash \phi$. Which, by the assumption about ϕ is equivalent to $\tau^n(k) \Vdash \phi$ (in the exact Kripke model) and by definition of χ also to $\tau^n(k) \Vdash \chi$. As χ is a $[\wedge,\vee,\neg]^n$ formula, $\tau^n(k) \Vdash \chi \;\Leftrightarrow\; k \Vdash \chi$, which proves $k \Vdash \phi$ to be equivalent to $k \Vdash \chi$. ⊣

3.4.1 The $[\wedge,\neg]$ fragments

We will prove that $[\wedge,\neg]$ is complete for models based on the simple frame of two connected worlds (and which will be called **2**). We will prove, that, as a consequence, the **IpL** fragment $[\wedge,\neg]$ is in fact the same as the $[\wedge,\neg]$ fragment of the three valued Heyting logic $\mathbf{H}_3$. The construction of the exact models for $\mathbf{H}_3^n$, the fragments of $\mathbf{H}_3$ with atoms restricted to the set $\{p_1,\ldots,p_n\}$, serves as an example to show that the technique of semantic types is also applicable in intermediate logics.

First however, we will compute the number of equivalence classes in $[\wedge,\neg]^n$, using the $Exm([\wedge,\vee,\neg]^n)$ from the previous subsection. Recall from fact 3.4.0.9 that the model $Exm([\wedge,\vee,\neg]^n)$ is isomorphic to the diagram of $[\wedge,\neg]^n$, without the bottom element.

3.4.1.12. THEOREM.

$$|Diag([\wedge, \neg]^n) = \sum_{k=0}^{n} \binom{n}{k} (2^{2^k} - 1) + 1.$$

Proof. If $S \subseteq \{p_1, \ldots, p_n\}$ and $|S| = k$, then there are 2^{n-k} sets U, in such a way that $S \subseteq U \subseteq \{p_1, \ldots, p_n\}$. Excluding the combination of S and $\langle S, \emptyset \rangle$, this implies that there are $2^{2^{n-k}} - 1$ semantic types t in $[\wedge, \vee, \neg]^n)$ with $j_0(t) = S$. As a consequence, we have

$$|Diag([\wedge, \neg]^n) = \sum_{k=0}^{n} \binom{n}{k} (2^{2^{n-k}} - 1) + 1.$$

Now use $\binom{n}{n-k} = \binom{n}{k}$ to obtain the formula in the theorem. $\dashv$

Let us first prove that $[\wedge, \neg]$ is complete for **2**-models, that is for models based on the frame **2**. In fact the theorem we will prove in the sequel is somewhat stronger and states that $[\wedge, \neg]^n$ is complete for the n-models based on **2**.

As a bridge between **IpL** models and **2**-models we first define *terminal models.*

3.4.1.13. DEFINITION. *If K is a finite* **IpL** *model K and $k, l \in K$ we call $\langle k, l \rangle$ a* terminal submodel *if l is a terminal node in K and $k < l$.*

Obviously a terminal submodel defined in K is a Kripke model in its own right as a submodel[2] of K.

3.4.1.14. LEMMA. *Let ϕ be a $[\wedge, \neg]$ formula, k a node in a finite* **IpL** *model K and $\langle k, l \rangle$ a terminal submodel in K. If $K \Vdash \phi$ then $\langle k, l \rangle \Vdash \phi$.*

Proof. By induction on the length of ϕ. If ϕ is atomic or a conjunction the proof is obvious. Note that in case $\phi = \neg\psi$, we may infer from $k \Vdash \neg\psi$ that the terminal node l will not force ψ. But the terminal node l above k in K and l in $\langle k, l \rangle$ force the same formulas. Hence $\langle k, l \rangle \Vdash \neg\psi$. $\dashv$

3.4.1.15. LEMMA. *Let ϕ be a $[\wedge, \neg]$ formula, k a node in a finite* **IpL** *model K. If $k \nVdash \phi$ then for some terminal submodel $\langle k, l \rangle$ in K, $\langle k, l \rangle \nVdash \phi$.*

Proof. By induction on the length of ϕ. The atomic and conjunction cases are easy. In case $\phi = \neg\psi$, we may infer from $k \nVdash \neg\psi$ that some terminal node $l \geq k$ must force ψ. Now any terminal model with this l will meet the condition from the lemma. $\dashv$

3.4.1.16. THEOREM. *The* **IpL** *fragment $[\wedge, \neg]$ is complete for* **2***-models.*

[2]Although not necessarily a generated submodel, as there may be an $m \in \uparrow k$ such that $m \notin \langle k, l \rangle$.

Proof. By combining lemma 3.4.1.14 and 3.4.1.15. $\dashv$

Let us briefly introduce the three valued Heyting logic, $\mathbf{H}_3$. The most concise definition of $\mathbf{H}_3$ would be: $\mathbf{H}_3$ is the logic of the **2**-models. If we use $\Vdash_2$ for forcing in **2**-models, $\phi \Vdash_2 \psi$ if for all k in a **2**-model $k \Vdash_2 \phi$ implies $k \Vdash_2 \psi$, and $\vdash_3$ for derivability in $\mathbf{H}_3$, then $\mathbf{H}_3$ being the logic of **2**-models comes down to:

$$\phi \vdash_3 \psi \quad \Leftrightarrow \quad \phi \Vdash_2 \psi.$$

An alternative, and more traditional, definition of $\mathbf{H}_3$ introduces $\vdash_3$ by truth tables for the connectives:

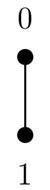
$\wedge$	f	$*$	t
f	f	f	f
$*$	f	$*$	$*$
t	f	$*$	t

$\vee$	f	$*$	t
f	f	$*$	t
$*$	$*$	$*$	t
t	t	t	t

$\rightarrow$	f	$*$	t
f	t	t	t
$*$	f	t	t
t	f	$*$	t

	$\neg$
f	t
$*$	f
t	f

It is left to the reader to check that these matrices correspond to the behavior of the connectives, according to the definition of forcing in **IpL** models, on the following **2**-model. Here the set $\{0,1\}$ represents the truth value **t**, $\{0\}$ the value $*$ and the empty set corresponds to the value **f**.



There are several alternative axiomatizations of the three valued Heyting logic.

3.4.1.17. FACT. $\mathbf{H}_3$ *can be axiomatized by adding one of the following formulas as an axiom to the axioms of* **IpL**.

1. $(p \leftrightarrow q) \vee (p \leftrightarrow r) \vee (p \leftrightarrow s) \vee (q \leftrightarrow r) \vee (q \leftrightarrow s) \vee (r \leftrightarrow s)$;
2. $p \vee (p \rightarrow q) \vee \neg q$;
3. $(((p \rightarrow (((q \rightarrow r) \rightarrow q) \rightarrow q)) \rightarrow p) \rightarrow p) \wedge ((p \rightarrow q) \vee (q \rightarrow p))$;
4. $((p \rightarrow q) \rightarrow r) \rightarrow (((s \rightarrow p) \rightarrow r) \rightarrow r)$.

The first of these axioms is Gödels formula expressing that there are only three truth values [Gödel 32]. The second is a simplified version of Hosoi's $p \vee \neg p \vee (p \rightarrow q) \vee (q \rightarrow r)$ in [Hosoi 66]. The first conjunct of 3 is the (once) iterated Peirce formula which is true exactly in the frames of depth less than two ([Gabbay 81]). The second conjunct is Dummett's axiom for the intermediate logic **LC**, the logic of linearly ordered frames[Gabbay 81]. In combination these formulas axiomatize the logic of linearly ordered frames of depth less than two, that is the frame **2** (and its subframe with only one world).

Formula 4 stems from Thomas [Thomas 62]. More details can be found in [Troelstra 65].

The definition of semantic types in $\mathbf{H}_3^n$ will not come as a surprise.

3.4.1.18. DEFINITION. *Let k be a node in a $\mathbf{2}$-model. The* semantic type *of k in $\mathbf{H}_3^n$ is defined by:*

$$\tau^n(k) = \begin{cases} \langle atom^n(k), \emptyset \rangle & \text{if} \quad \forall l > k. atom^n(k) = atom^n(l) \\ \\ \langle atom^n(k), \{\tau^n(l) \mid k < l\} \rangle & \text{otherwise.} \end{cases}$$

The order of semantic types t and t' in $\mathbf{H}_3^n$ is defined by

$$t \preceq t' \quad \Leftrightarrow \quad t = t' \text{ or } t' \in j_1(t).$$

Define the type, *$\phi^n(k)$, of k in $\mathbf{H}_3^n$ by:*

$$\phi^n(k) = \begin{cases} \phi^n_{\mathbf{CpL}}(k) & \text{if} \quad j_1(\tau^n(k)) = \emptyset \\ \\ \bigwedge j_0(\tau^n(k)) \wedge \\ \bigwedge \{\neg\neg p \mid p \in j_0(\tau^n(l))\} \wedge \\ \bigwedge \{\neg p \mid p \in \{p_1, \ldots, p_n\} \setminus j_0(\tau^n(l))\} \wedge \\ \bigwedge \{p \leftrightarrow q \mid p, q \in j_0(\tau^n(l)) \setminus j_0(\tau^n(k))\} & \text{if} \quad j_1(\tau^n(k)) = \{\tau^n(l)\}. \end{cases}$$

Observe that in particular $\tau^n(k) = \langle atom^n(k), \emptyset \rangle$ if k is a terminal node. Moreover, if t is a semantic type in $\mathbf{H}_3^n$ and $j_1(t) \neq \emptyset$, then $j_1(t) = \{t'\}$ and $j_1(t') = \emptyset$.

Observe also, that if $k \leq l$ in a $\mathbf{2}$-model, then $\tau^n(k) \preceq \tau^n(l)$.

As can be verified easily, the definition of $\phi^n(k)$ assures that $k \Vdash \phi^n(k)$ for k a node in a $\mathbf{2}$-model. Note that if k is a terminal node then $\phi^n(k) \equiv \phi^n_{\mathbf{CpL}}(k)$.

Now we are ready to prove, like we did in lemma 3.4.0.5 for $[\wedge, \vee, \neg]^n$, that the types and semantic types introduced for $\mathbf{H}_3^n$ behave like one would expect. In the sequel of this subsection we will use $Th^n(k)$ for the theory of formulas in $\mathbf{H}_3^n$ forced by k.

3.4.1.19. LEMMA. *Let k and l be nodes in $\mathbf{2}$-models. Then the following statements are equivalent:*

1. *$l \Vdash \phi^n(k)$;*
2. *$\tau^n(k) \preceq \tau^n(l)$;*
3. *$Th^n(k) \subseteq Th^n(l)$;*
4. *$\phi^n(l) \vdash \phi^n(k)$.*

Proof. We will prove $1 \Rightarrow 2 \Rightarrow 3 \Rightarrow 4 \Rightarrow 1$.

$1 \Rightarrow 2$: We have to prove that either $\tau^n(k) = \tau^n(l)$ or l is a terminal node and $j_1(\tau^n(k)) = \{\tau^n(l)\}$.

In case l and k are both terminal nodes $\phi^n(k)$ is a $\mathbf{CpL}^n$ type and obviously $l \Vdash \phi^n(k)$ implies $\tau^n(k) = \tau^n(l)$. If l is a terminal node and k is not, let k' be the terminal node above k. We will prove $\tau^n(l) = \tau^n(k')$. For $p \in atom^n(k')$ infer from the definition of the type of k that $\phi^n(k) \vdash \neg\neg p$. As $l \Vdash \phi^n(k)$ this assures us that $p \in atom^n(l)$. Hence $atom^n(k') \subseteq atom^n(l)$. Likewise, if $p \notin atom^n(k')$ then $\phi^n(k) \vdash \neg p$ and hence $p \notin atom^n(l)$. Which proves $atom^n(k') = atom^n(l)$ and, as both are terminal nodes, $\tau^n(k) = \tau^n(l)$. Note that if k is a terminal node

$l \Vdash \phi^n_{\mathbf{CpL}}(k)$ obviously implies that $j_1(\tau^n(l)) = \emptyset$ and hence $\tau^n(k) = \tau^n(l)$. So suppose both $j_1(\tau^n(l)) \neq \emptyset$ and $j_1(\tau^n(k)) \neq \emptyset$. Then there is an $l' > l$ such that $l' \Vdash \phi^n(k)$. Again we may infer that $\tau^n(k') = \tau^n(l')$ for the terminal node k' above k. Clearly $atom^n(k) \subseteq atom^n(l)$ and to prove $atom^n(l)$ to be a subset of $atom^n(k)$, let $p \in atom^n(l)$. Now either $p \in atom^n(k)$, or $p \in atom^n(k') \setminus atom^n(k)$ or p is not in $atom^n(k')$. In the first case we are ready and in the third case $\phi^n(k) \vdash \neg p$, contradicting $p \in atom^n(l)$. If $p \in atom^n(k') \setminus atom^n(k)$, note that, as $j_1(\tau^n(l)) \neq \emptyset$ there is a $q \in atom^n(l') \setminus atom^n(l)$. As $atom^n(l') = atom^n(k')$ we will have $\phi^n(k) \vdash p \to q$, contradicting $p \in atom^n(l)$.

2 $\Rightarrow$ 3: Assume $\tau^n(k) \preceq \tau^n(l)$. If $j_1(\tau^n(k)) = \emptyset$, then obviously $k \rightleftarrows l$ and $Th^n(k) = Th^n(l)$. On the other hand, if $j_1(\tau^n(k)) = \{\tau^n(k')\}$, then either $\tau^n(l) = \tau^n(k')$ or $j_1(\tau^n(l)) = \{\tau^n(l')\}$, $\tau^n(l') = \tau^n(k')$ and $atom^n(k) = atom^n(l)$. As k' is a terminal node, from $\tau^n(l) = \tau^n(k')$ we may conclude $l \rightleftarrows k'$ and hence $Th^n(k) \subset Th^n(k') = Th^n(l)$. In case $j_1(\tau^n(l)) = \{\tau^n(l')\}$, we conclude from $\tau^n(l') = \tau^n(k')$, that $l' \rightleftarrows k'$. As also $atom^n(k) = atom^n(l)$, we may infer that $k \rightleftarrows l$ and hence $Th^n(k) = Th^n(l)$.

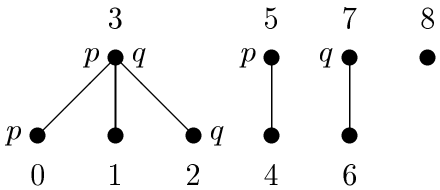
3 $\Rightarrow$ 4: As in the proof of theorem 3.4.0.5 we conclude from the previous steps that in general $\phi^n(m)$ is an axiom of $Th^n(m)$. Hence from $Th^n(k) \subseteq Th^n(l)$ we conclude that $\phi^n(l) \vdash \phi^n(k)$.

4 $\Rightarrow$ 1: As observed earlier $l \Vdash \phi^n(l)$ is a simple consequence of definition 3.4.1.18. Hence trivially, $\phi^n(l) \vdash \phi^n(k)$ implies $l \Vdash \phi^n(k)$. $\dashv$

Now define $Exm(\mathbf{H}^n_3)$ as the ordered set of semantic types in $\mathbf{H}^n_3$. Obviously $Exm(\mathbf{H}^n_3)$ is a Kripke model if we take $atom^n(t) = j_0(t)$ as its valuation. Note that $Exm(\mathbf{H}^n_3)$ will again be a **2**-model.

As $Exm(\mathbf{H}^n_3)$ realizes all semantic types in $\mathbf{H}^n_3$ with lemma 3.4.1.19 one easily proves that the model is complete for $\mathbf{H}^n_3$. Moreover, for every node $k \in Exm(\mathbf{H}^n_3)$ we have a type formula $\phi^n(k)$ such that $[\![\phi^n(k)]\!] = \uparrow k$. As closed subsets in $Exm(\mathbf{H}^n_3)$ correspond to disjunctions of these type formulas we may conclude that $Exm(\mathbf{H}^n_3)$ is the exact Kripke model of $\mathbf{H}^n_3$.

3.4.1.20. THEOREM. *The model* $Exm(\mathbf{H}_3)$ *defined above is the exact Kripke model of* $\mathbf{H}^n_3$.



13. FIGURE. *The exact Kripke model of* $\mathbf{H}^2_3$.

The irreducible formulas in $\mathbf{H}^2_3$ are:

0. $p \wedge \neg\neg q$	3. $p \wedge q$	6. $\neg p \wedge \neg\neg q$
1. $\neg\neg p \wedge (p \leftrightarrow q)$	4. $\neg\neg p \wedge \neg q$	7. $\neg p \wedge q$
2. $\neg\neg p \wedge q$	5. $p \wedge \neg q$	8. $\neg p \wedge \neg q$

Note that, in contrast to in **IpL**, not all $[\wedge, \neg]^2$ formulas are irreducible in $\mathbf{H}_3$.

For example (as can be proved using the exact Kripke model of $\mathbf{H}_3^2$): $\neg\neg p \wedge \neg\neg q \equiv (\neg\neg p \wedge (p \leftrightarrow q)) \vee (p \wedge \neg\neg q) \vee (\neg\neg p \wedge q)$.

The model above has been used to calculate the diagram of $\mathbf{H}_3^2$. A listing of all 162 equivalence classes can be found in appendix B.2.

From the structure of the exact Kripke model of $\mathbf{H}_3^n$ one can calculate the number of elements in $Exm(\mathbf{H}_3)$ as $\sum_{k=0}^{n} 2^k.\binom{n}{k}$ and the number of classes in $Diag(\mathbf{H}_3^n)$ as:

$$\prod_{k=0}^{n} (2^{2^k-1} + 1)^{\binom{n}{k}}.$$

3.4.2 The $[\wedge, \neg\neg]$ fragments

The $[\wedge, \neg\neg]$ fragments have rather simple and regular diagrams. The expressive power of these fragments is too limited to be of very much interest, but each $[\wedge, \neg\neg]^n$ fragment 'almost' has an exact model. For formulas in $[\wedge, \neg\neg]^n$ we have an obvious normal form.

3.4.2.21. FACT. (The $[\wedge, \neg\neg]^n$ normal form) *Each formula in $[\wedge, \neg\neg]^n$ is equivalent to a formula of the form $\bigwedge P \wedge \bigwedge\{\neg\neg q \mid q \in Q\}$ where both P and Q are subsets of $\{p_1, \ldots, p_n\}$, $P \cup Q \neq \emptyset$ and $P \cap Q = \emptyset$.*

To characterize the semantic types for $[\wedge, \neg\neg]^n$, let us introduce a special type of **IpL** models, *n-maximal models* .

3.4.2.22. DEFINITION. *A finite n-model K is called n-maximal if each $k \in K$ forces at least $n-1$ atoms.*

3.4.2.23. THEOREM. *If ϕ and ψ formulas in $[\wedge, \neg\neg]^n$ such that $\phi \nvdash \psi$ then there is a node k in an n-maximal model such that $k \Vdash \phi$ and $k \nVdash \psi$.*

Proof. Let $\bigwedge P \wedge \bigwedge\{\neg\neg q \mid q \in Q\}$ be the normal form of ϕ and $\bigwedge R \wedge \bigwedge\{\neg\neg q \mid q \in S\}$ the normal form of ψ. From $\phi \nvdash \psi$ we may infer that either there is an $r \in R$ such that $r \notin P$ or there is an $s \in S$ such that $s \notin P \cup Q$.

In the first case, let $atom^n(k)$ contain all atoms but r and $k < l$ such that $atom^n(l) = \{p_1, \ldots, p_n\}$. Obviously $k \Vdash \phi$ but $k \nVdash \psi$.

In the second case, let k be a node forcing all atoms in $\{p_1, \ldots, p_n\}$ except s. Let k have two successors l_0 and l_1, with $atom^n(l_0) = \{p_1, \ldots, p_n\}$ and $atom^n(l_1) = atom^n(k)$. Again k will force ϕ but not ψ. $\dashv$

3.4.2.24. COROLLARY. *The fragment $[\wedge, \neg\neg]^n$ is complete for n-maximal models.*

Recall the definition of $Ter(k)$ from definition 3.4.0.3 as the set of all terminal nodes above the node k. The proof of theorem 3.4.2.23 motivates the following definition of semantic type (in $[\wedge, \neg\neg]^n$) for a node in an n-maximal model.

3.4.2.25. DEFINITION. *Let k be a node in an n-maximal model. Then $\tau^n(k)$, the semantic type of k in $[\wedge, \neg\neg]^n$ is defined by:*

$$\tau^n(k) = \begin{cases} \langle atom^n(k), \emptyset \rangle & \text{if} \quad \forall l > k.\, atom^n(l) = atom^n(k) \\ \langle atom^n(k), \{\tau^n(l) \mid l \in Ter(k)\} \rangle & \text{otherwise.} \end{cases}$$

For semantic types t and t' in $[\wedge, \neg\neg]^n$ define:

$$t \preceq t' \quad \Leftrightarrow \quad t = t' \text{ or } t' \in j_1(t) \text{ or } (j_0(t) \subseteq j_0(t') \text{ and } \emptyset \neq j_1(t') \subseteq j_1(t)).$$

3.4.2.26. DEFINITION. *A node k in an n-maximal model is called a* proper *node, if $j_1(\tau^n(k)) \neq \emptyset$.*

Inspection of the proof of the theorem 3.4.2.23 reveals the following fact.

3.4.2.27. FACT. *If ϕ and ψ formulas in $[\wedge, \neg\neg]^n$ such that $\phi \nvdash \psi$ then there is a proper node k in an n-maximal model such that $k \Vdash \phi$ and $k \nVdash \psi$.*

Note that the ordered set of semantic types of proper nodes for $[\wedge, \neg\neg]^n$, n disjoint **2**-models, is not a Kripke model realizing all the semantic types in $[\wedge, \neg\neg]^n$. By adding terminal nodes with semantic type $\langle Q, \emptyset \rangle$ where $|Q| \geq n-1$, the ordered set of types becomes an n-maximal model.

3.4.2.28. DEFINITION. *The model $Umod([\wedge, \neg\neg]^n)$ is the ordered set of semantic types in $[\wedge, \neg\neg]^n$.*

To prove that $Umod([\wedge, \neg\neg]^n)$ is a universal model for $[\wedge, \neg\neg]^n$ we will show that it is the Kripke completion of the exact model of the fragment $[\wedge, \neg\neg, \top]^n$, that is the fragment $[\wedge, \neg\neg]^n$ with the formula $\top$ added.

In the proof we will need the (formula) types in $[\wedge, \neg\neg]^n$.

3.4.2.29. DEFINITION. *If k a proper node in an n-maximal Kripke model, then $\phi^n(k)$, the* type *of k in $[\wedge, \neg\neg]^n$, is defined as:*

$$\phi^n(k) = \bigwedge j_0(\tau^n(k)) \wedge \bigwedge \{\neg\neg q \mid q \in \bigcap \{j_0(t) \mid t \in j_1(\tau^n(k))\}\}.$$

3.4.2.30. THEOREM. *The model $Umod([\wedge, \neg\neg]^n)$ defined above is a universal model for $[\wedge, \neg\neg]^n$ and the ordered set of semantic types in $[\wedge, \neg\neg]^n$ is an exact model for $[\wedge, \neg\neg, \top]^n$.*

Proof. $Umod([\wedge, \neg\neg]^n)$ clearly is complete for $[\wedge, \neg\neg]^n$ and minimal in realizing all the semantic types of proper nodes in $[\wedge, \neg\neg]^n$.

Hence our main task will be to prove that we realy need all semantic types in $[\wedge, \neg\neg]^n$. First note that if k is a node in the intended universal model with a semantic type in $[\wedge, \neg\neg]^n$, then $k \Vdash \phi^n(k)$. From the definition of $\phi^n(k)$ it is also clear that in *Umod*($[\wedge,\neg\neg]^n$) it is true that $l \Vdash \phi^n(k)$ iff $k \leq l$. Hence, for k such that $\tau^n(k)$ is a semantic type in $[\wedge,\neg\neg]^n$ we have $[\![\phi^n(k)]\!] = \uparrow k$.

Suppose $k_1,\ldots,k_m$ is a close subset in the submodel of the proper nodes in *Umod*($[\wedge,\neg\neg]^n$). Let ϕ be the formula

$$\phi = \bigwedge \bigcap_{i=1}^{m} j_0(\tau^n(k_i)) \wedge \bigwedge\{\neg\neg p \mid p \in \bigcap_{i=1}^{m} \bigcap\{j_0(t) \mid t \in j_1(\tau^n(k_i))\}\}.$$

Note that if $k_1,\ldots,k_m$ is the set of all proper nodes in *Umod*($[\wedge,\neg\neg]^n$), then $\phi = \top$ (which is not a $[\wedge,\neg\neg]^n$ formula).

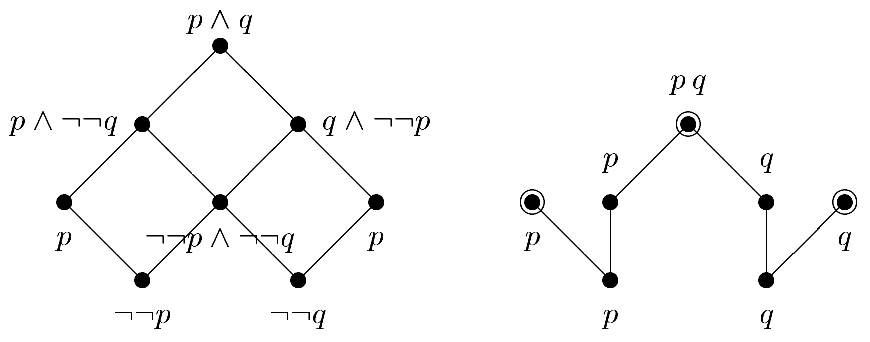
To prove that $[\![\phi]\!] = \uparrow\{k_1,\ldots,k_n\}$, let $k \in \{k_1,\ldots,k_n\}$. Then $\bigcap_{i=1}^{m} j_0(\tau^n(k_i)) \subseteq j_0(\tau^n(k))$ and $\bigcap_{i=1}^{m} \bigcap\{j_0(t) \mid t \in j_1(\tau^n(k_i))\} \subseteq \bigcap\{j_0(t) \mid t \in j_1(\tau^n(k))\}$. From which we may infer that $k \Vdash \phi$.

On the other hand, if $k \Vdash \phi$, suppose k is a terminal node in *Umod*($[\wedge,\neg\neg]^n$). If $atom^n(k) = \{p_1,\ldots,p_n\}$, then clearly $k_i > k$ for all k_i. If $|atom^n(k)| = n-1$, then there is exactly one l in *Umod*($[\wedge,\neg\neg]^n$) such that $l < k$. Let $q \in \{p_1,\ldots,p_n\} \setminus atom^n(k)$, then $l \in \{k_1,\ldots,k_m\}$ iff $\phi \nvdash \neg\neg q$. Hence, from $k \Vdash \phi$ we conclude $\phi \nvdash \neg\neg q$ and hence $k \in \uparrow\{k_1,\ldots,k_m\}$. In case k is a proper node of *Umod*($[\wedge,\neg\neg]^n$), for $q \in \{p_1,\ldots,p_n\} \setminus atom^n(k)$ we have $\phi \nvdash q \quad \Leftrightarrow \quad k \in \{k_1,\ldots,k_m\}$. From $k \Vdash \phi$, we infer that $\phi \nvdash q$ and hence $k \in \{k_1,\ldots,k_m\}$.

Hence, we proved that the ordered set of semantic types in $[\wedge,\neg\neg]^n$ is an exact model for $[\wedge,\neg\neg,\top]^n$. $\dashv$

3.4.2.31. COROLLARY. *The exact model of $[\wedge,\neg\neg,\top]^n$ is isomorphic with n disjoint copies of* **2** *(disregarding the valuation in Umod($[\wedge,\neg\neg]^n$)).*

Hence the fragment $[\wedge,\neg\neg,\top]^n$ has $3^n - 1$ equivalence classes.



14. FIGURE. *The diagram of $[\wedge,\neg\neg]^2$ and the Kripke completion of its exact model (with the added terminal nodes encircled).*

3.4.3 The $[\wedge, \vee, \neg\neg]$ fragments

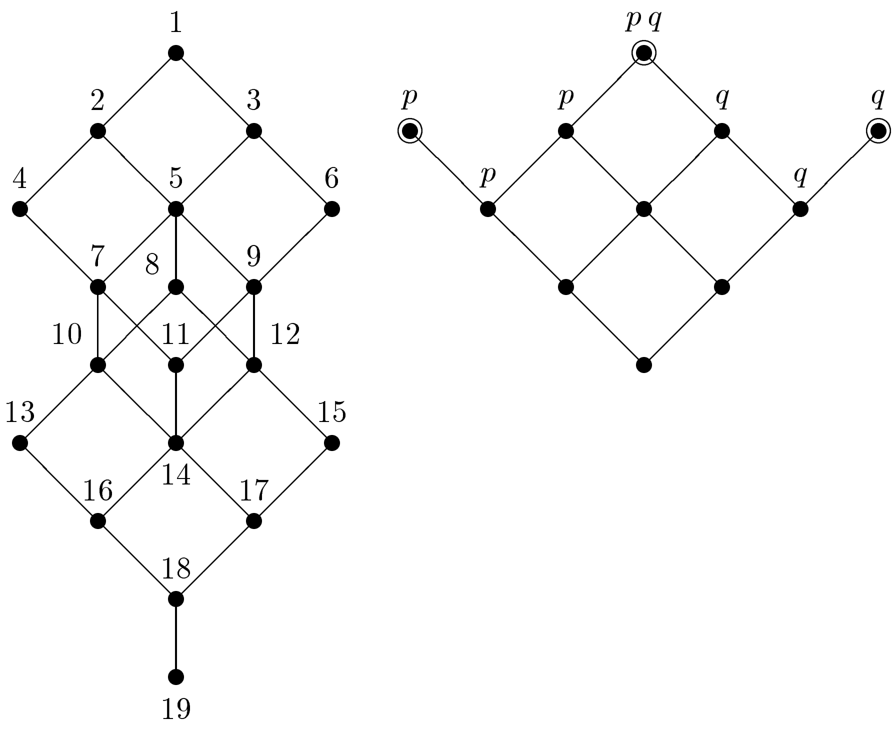
The construction of the exact model of $[\wedge,\vee,\neg\neg]^n$ from the irreducible formulas in this fragment is rather straightforward.

3.4.3.32. Lemma. *The irreducible formulas in* $[\wedge, \vee, \neg\neg]^n$ *are (modulo logical equivalence) of the form* $\bigwedge Q \wedge \neg\neg\psi$*, where* Q *is some subset of* $\{p_1, \ldots, p_n\}$ *and* ψ *is some formula in* $[\wedge, \vee]^n$*.*

Proof. If $\phi = \bigwedge Q \wedge \neg\neg\psi$ and $\phi \nvdash \bot$, then ϕ is $\vee$-irreducible by 3.2.0.3.4. If $\phi \in [\wedge, \vee, \neg\neg]^n$, it is not difficult to prove ϕ to be equivalent to a disjunction of formulas of the form $\bigwedge Q \wedge \neg\neg\psi$. Hence, if ϕ is irreducible, ϕ is equivalent to a formula of the form $\bigwedge Q \wedge \neg\neg\psi$. ⊣

3.4.3.33. Corollary. (The $[\wedge, \vee, \neg\neg]^n$ normal form) *Every formula in* $[\wedge, \vee, \neg\neg]$ *is equivalent to a disjunction of formulas of the form* $\bigwedge Q \wedge \neg\neg\psi$ *where* Q *is some subset of* $\{p_1, \ldots, p_n\}$ *and* ψ *is a formula in* $[\wedge, \vee]^n$*.*

With exception of the fragment with only one atom, these exact models are not exact Kripke models, as can be seen from the example of $[\wedge, \vee, \neg\neg]^2$.



15. Figure. *The diagram of* $[\wedge, \vee, \neg\neg]^2$ *and the Kripke completion of its exact model (the encircled nodes have been added).*

The formulas in the diagram of $[\wedge, \vee, \neg\neg]^2$:

1. $p \wedge q$	8. $\neg\neg(p \wedge q)$	15. $\neg\neg q$
2. $p \wedge \neg\neg q$	9. $(p \vee q) \wedge \neg\neg q$	16. $\neg\neg p \vee q$
3. $\neg\neg p \wedge q$	10. $\neg\neg p \wedge (p \vee \neg\neg q)$	17. $p \vee \neg\neg q$
4. p	11. $p \vee q$	18. $\neg\neg p \vee \neg\neg q$
5. $\neg\neg(p \wedge q) \wedge (p \vee q)$	12. $(\neg\neg p \vee q) \wedge \neg\neg q$	19. $\neg\neg(p \vee q)$
6. q	13. $\neg\neg p$	
7. $\neg\neg p \wedge (p \vee q)$	14. $(p \vee \neg\neg q) \wedge (\neg\neg p \vee q)$	

To find the semantic types in $[\wedge, \vee, \neg\neg]^n$ the normal form of the irreducible formulas suggests the following definitions.

3.4.3.34. DEFINITION. *A finite IpL model K is called a* proper $[\wedge, \vee, \neg\neg]^n$ model *if for no terminal node l it is true that $atom^n(l) = \emptyset$ and for every $k \in K$ which is not a terminal node, there is a terminal $l > k$ with $atom^n(l) = \{p_1, \ldots, p_n\}$.*

3.4.3.35. DEFINITION. *For k a node in a proper $[\wedge, \vee, \neg\neg]^n$ model define $\tau^n(k)$, the* semantic type *of k in $[\wedge, \vee, \neg\neg]^n$, as:*

$$\tau^n(k) = \begin{cases} \langle atom^n(k), \emptyset\rangle & \text{if} \quad \forall l > k.\, atom^n(l) = atom^n(k) \\ \\ \langle atom^n(k), \{\tau^n(l) \mid l \in Ter(k)\}\rangle & \text{otherwise} \end{cases}$$

For semantic types t and t' in $[\wedge, \neg\neg]^n$ define:

$$t \preceq t' \quad \Leftrightarrow \quad t = t' \text{ or } t' \in j_1(t) \text{ or } (j_0(t) \subseteq j_0(t') \text{ and } \emptyset \neq j_1(t') \subseteq j_1(t)).$$

Define $\phi^n(k)$, the type *of k in $[\wedge, \vee, \neg\neg]^n$, as:*

$$\phi^n(k) = \bigwedge j_0(\tau^n(k)) \wedge \neg\neg\bigvee\{\bigwedge j_0(t) \mid t \in j_1(\tau^n(k))\}$$

Observe that for each k in a proper $[\wedge, \vee, \neg\neg]^n$ model which is not a terminal node, we have $k \Vdash \phi^n(k)$.

To prove that each irreducible formula of $[\wedge, \vee, \neg\neg]^n$ is a $\phi^n(k)$ for some non-terminal node k in a proper $[\wedge, \vee, \neg\neg]^n$ model, first note that in the normal form of irreducible formulas in $[\wedge, \vee, \neg\neg]^n$ the part in the scope of $\neg\neg$ is a formula of $[\wedge, \vee]$ and hence needs for its realization terminal nodes l with $atom^n(l) \neq \emptyset$. A second observation we need is that for $\psi \in [\wedge, \vee]^n$ it is always true that $\neg\neg(\psi \vee \bigwedge\{p_1, \ldots, p_n\}) \equiv \neg\neg\psi$.

So we may infer that $[\wedge, \vee, \neg\neg]^n$ is complete for proper $[\wedge, \vee, \neg\neg]^n$ models, as every irreducible formula in the fragment can be realized in one of these models.

3.4.3.36. FACT. *The fragment $[\wedge, \vee, \neg\neg]^n$ is complete for proper $[\wedge, \vee, \neg\neg]^n$ models.*

The ordered set of semantic types in $[\wedge, \vee, \neg\neg]^n$ will provide us with an exact model of $[\wedge, \vee, \neg\neg]^n$.

3.4.3.37. DEFINITION. *Let* $\mathrm{Umod}([\wedge, \vee, \neg\neg]^n)$ *be the Kripke model constructed from the ordered set of semantic types $\langle Q, T\rangle$, such that $Q \subseteq \{p_1, \ldots, p_n\}$ and T a set of semantic types $\langle U, \emptyset\rangle$, such that $\langle\{p_1, \ldots, p_n\}\emptyset\rangle \in T$. The valuation in* $\mathrm{Umod}([\wedge, \vee, \neg\neg]^n)$ *is defined by $atom^n(t) = j_0(t)$.*

Obviously this model is a proper $[\wedge, \vee, \neg\neg]^n$ model realizing all semantic types in the fragment and hence it is complete for $[\wedge, \vee, \neg\neg]^n$. However, as an exact model *Umod*$([\wedge, \vee, \neg\neg]^n)$ is too large. More precisely we do not need the terminal nodes in this model, as observed earlier. Note that the semantic type $\langle\{p_1, \ldots, p_n\}, \emptyset\rangle$, corresponding to the type $\bigwedge\{p_1, \ldots, p_n\}$, will only be realized in the model as the type of a terminal node. However as this type acts as a bottom element in the diagram of the fragment, it is not needed in the exact model as it will correspond to the empty set.

3.4.3.38. THEOREM. *The model* Umod$([\wedge, \vee, \neg\neg]^n)$ *without its terminal nodes is the exact model of* $[\wedge, \vee, \neg\neg]^n$.

Proof. As we have seen, every non-terminal element $\langle Q, T\rangle$ in *Umod*$([\wedge, \vee, \neg\neg]^n)$ corresponds to a type $\bigwedge Q \wedge \neg\neg\bigvee\{\bigwedge j_0(t) \mid t \in T\}$ and every irreducible formula in $[\wedge, \vee, \neg\neg]^n$ is equivalent to such a type. The only thing we still have to prove is that different semantic types indeed have different type formulas. This we may infer from the fact that for t and t' semantic types in $[\wedge, \vee, \neg\neg]^n$ and ϕ_t and $\phi_{t'}$ the corresponding type formulas it is true that $t \preceq t' \quad \Leftrightarrow \quad \phi_{t'} \vdash \phi_t$. The proof of this last fact is straightforward from the definitions and is left to the industrious reader. $\dashv$

3.4.4 The $[\vee, \neg]$ fragments

As was already mentioned before, a fragment $[\vee, \neg]^n$ has an exact model. For $n > 1$ this is not an exact Kripke model, as we will see.

The fragment $[\vee, \neg]^n$ is a subfragment of $[\wedge, \vee, \neg]^n$, for which we already defined semantic types and constructed an exact Kripke model. As the irreducible formulas in $[\vee, \neg]^n$ are also irreducible in $[\wedge, \vee, \neg]^n$ we have already met the (semantic) types in $[\vee, \neg]^n$: those types in $[\wedge, \vee, \neg]^n$ which are equivalent to a formula in $[\vee, \neg]^n$ (and the corresponding semantic types).

Of course the only irreducible formulas in $[\vee, \neg]^n$ are the atomic formulas and the negations. Note also that conjunctions of negations are equivalent to negations of disjunctions (i.e. $\neg p \wedge \neg q \equiv \neg(p \vee q)$), and thus are part of the fragment.

Recall that semantic types in $[\wedge, \vee, \neg]^n$ are of the form:

$$\tau^n(k) = \begin{cases} \langle atom^n(k), \emptyset\rangle & \text{if} \quad \forall l > k.\, atom^n(l) = atom^n(k) \\ \\ \langle atom^n(k), \{\tau^n(l) \mid l \in Ter(k)\}\rangle & \text{otherwise} \end{cases}$$

The corresponding type $\phi^n(k)$ was defined as:

$$\phi^n(k) = \begin{cases} \phi^n_{CpL}(k) & \text{if } \forall l > k.\, atom^n(l) = atom^n(k) \\ \\ \bigwedge j_0(\tau^n(k)) \wedge \\ \neg\neg\bigvee\{\phi^n_{CpL}(l) \mid \tau^n(l) \in j_1(\tau^n(k))\} & \text{otherwise} \end{cases}$$

If a formula type $\phi^n(k)$ in $[\wedge, \vee, \neg]^n$ corresponds to a formula in $[\vee, \neg]^n$ then

1. $atom^n(k)$ is either empty or a singleton;
2. k if $atom^n(k) \neq \emptyset$ and $n > 1$ then k is not a terminal node.

Note that in the first case, where $\phi^n(k)$ is equivalent to an atomic formula p, we have $\bigvee\{\phi^n_{CpL}(l) \mid \tau^n(l) \in j_1(\tau^n(k))\} \equiv p$. Hence, for all $Q \in \{p_1, \ldots, p_n\}$ such that $p \in Q$, there is a $t \in j_1(\tau^n(k))$ with $j_0(t) = Q$. Otherwise, we would have $k \Vdash \neg\phi^n_Q$, which contradicts $\phi^n(k) \equiv p$, as $p \nvdash \neg\phi^n_Q$.

As it is not difficult to see that every formula in $[\wedge, \vee, \neg]^1$ is equivalent to a formula in $[\vee, \neg]^1$ (see figure 9), the two fragments have the same diagram. In the sequel of this subsection we will assume $n > 1$, without making this exception explicit every time we should.

The observations above inspired the definition of a semantic type in $[\vee, \neg]^n$.

3.4.4.39. Definition. *Let k be a node in a finite IpL model. Then k is a* proper $[\vee, \neg]^n$ node *if $atom^n(k) = \emptyset$ or $atom^n(k) = \{p\}$ for some $p \in \{p_1, \ldots, p_n\}$ and for every $Q \subseteq \{p_1, \ldots, p_n\}$ such that $p \in Q$ there is an $l \in Ter(k)$ with $atom^n(l) = Q$.*

If k is a proper $[\vee, \neg]^n$ node, then $\tau^n(k)$, the semantic type *of k in $[\wedge, \vee, \neg]^n$ is the semantic type of k in $[\vee, \neg]^n$ and $\phi^n(k)$, the type formula of k in $[\wedge, \vee, \neg]^n$, is the type formula of k in $[\vee, \neg]^n$.*

Let $Th^n(k)$ in the sequel of this subsection be the $[\vee, \neg]^n$ theory of k, $Th^n(k) = \{\phi \in [\vee, \neg]^n \mid k \Vdash \phi\}$.

To see that $\phi^n(k)$ is a formula in $[\vee, \neg]^n$ note that either $atom^n(k) = \{p\}$ for some atom p or else $\phi^n(k)$ is a negation.

3.4.4.40. Lemma. *If k and l are nodes in finite Kripke models and k and l have semantic types in $[\vee, \neg]^n$, then:*

$$\tau^n(k) \preceq \tau^n(l) \quad \Leftrightarrow \quad Th^n(k) \subseteq Th^n(l) \quad \Leftrightarrow \quad l \Vdash \phi^n(k)$$

Proof. The lemma is an application of lemma 3.4.0.5, in the special case where k and l have semantic types in $[\vee, \neg]^n$ and the theories $Th^n(k), Th^n(l)$ and the formula $\phi^n(k)$ are in $[\vee, \neg]^n$. $\dashv$

By ordering the semantic types in $[\vee, \neg]^n$, adding the terminal nodes which have no type in $[\vee, \neg]^n$, we get a Kripke completion of the exact model of $[\vee, \neg]^n$ as we will see.

3.4.4.41. Definition. *Let* $\mathrm{Umod}([\vee, \neg]^n) = \langle T, \preceq, j_0\rangle$ *be the Kripke model constructed from the set T of both semantic types in $[\vee, \neg]^n$ and types of the form $\langle Q, \emptyset\rangle$, where Q a nonempty subset of $\{p_1, \ldots, p_n\}$. The order relation between these types is defined as*

$$t \preceq t' \quad \Leftrightarrow \quad t = t' \text{ or } t' \in j_1(t) \text{ or } (j_0(t) \subseteq j_0(t') \text{ and } \emptyset \neq j_1(t') \subseteq j_1(t)).$$

The valuation in $\mathrm{Umod}([\vee, \neg]^n)$ *is defined by $atom^n(t) = j_0(t)$.*

From the definitions and observations above the following fact is a simple consequence.

3.4.4.42. FACT. *The model* Umod($[\vee, \neg]^n$) *realizes each semantic type in* $[\vee, \neg]^n$.

3.4.4.43. THEOREM. *The model* Umod($[\vee, \neg]^n$) *is a universal model for the fragment* $[\vee, \neg]^n$ *and the ordered set of semantic types in* $[\vee, \neg]^n$ *is the exact model of* $[\vee, \neg]^n$.

Proof. The irreducible formula classes of $[\vee, \neg]^n$ correspond exactly to the semantic types in $[\vee, \neg]^n$. If X is an upwardly closed subset of types $\{\tau^n(k_1), \ldots, \tau^n(k_m)\}$, this X will correspond to the formula $\bigvee\{\phi^n(k_1), \ldots, \phi^n(k_m)\}$ (where $\bigvee\emptyset = \bot$).

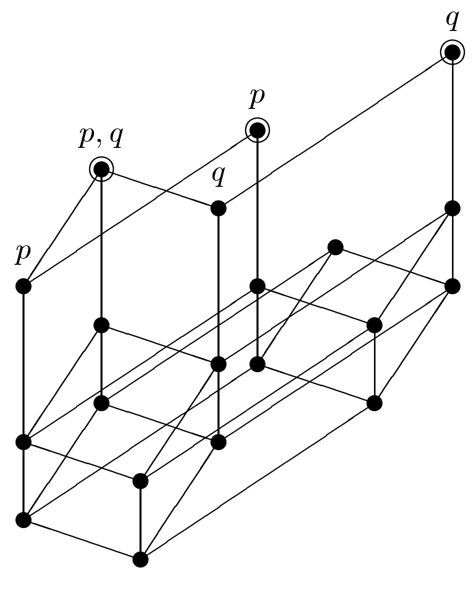
To realize the semantic types in $[\vee, \neg]^n$ one obviously needs the terminal nodes forcing non-empty sets of atoms. As these are the only elements added in *Umod*($[\vee, \neg]^n$), this model is a minimal Kripke completion of the exact model. ⊣

The structure of the exact model of $[\vee, \neg]^n$ might also be described as that of the ordered set of all non-contradictory negations where the atomic formulas are added.

The set of non-contradictory negations is isomorphic to the diagram of the classical diagram $[\vee, \neg]^n_{CpL}$ without the tautology and hence also with the 2^n-dimensional hypercube without a top.

Hence the universal model of $[\vee, \neg]^n$ will be an n-dimensional hypercube without a top, where at n corners there have been added nodes that force just one atom and which have been connected to the $2^n - 1$ terminal nodes outside the exact model.

As an illustration, figure 16 shows the universal model of $[\vee, \neg]^2$.



16. FIGURE. *The universal model of* $[\vee, \neg]^2$. *(The encircled nodes have been added to the exact model.)*

Using this model one can compute the diagram of $[\vee, \neg]^2$ which has 385 equivalent classes.

3.4.5 The $[\vee,\neg\neg]$ fragments

With exception of $[\vee,\neg\neg]^1$, the fragments $[\vee,\neg\neg]^n$ will not have an exact model. Note that the minimal elements in the diagram of $[\vee,\neg\neg]^n$ are the atomic formulas and for $n > 1$ there will not be a bottom in $Diag([\vee,\neg\neg]^n)$. Hence the diagram of $[\vee,\neg\neg]^n$ is not a lattice.

But by adding $\bot$ to $[\vee,\neg\neg]^n$ we will have a fragment with an exact Kripke model as we will see.

The fragment $[\vee,\neg\neg]^1$ has a simple diagram (two classes, p and $\neg\neg p$) and has an exact model (with $\neg\neg(p)$ as its only element and p corresponding to the empty set) which is not an exact Kripke model. In the sequel of this subsection we will assume $n > 1$.

There is a simple normal form in $[\vee,\neg\neg]^n$ which we will use in the construction of this exact Kripke model of $[\vee,\neg\neg,\bot]^n$.

3.4.5.44. FACT. (The $[\vee,\neg\neg]^n$ normal form) *Every formula in $[\vee,\neg\neg]^n$ is equivalent to a disjunction of formulas that are either atomic or of the form $\neg\neg\psi$, where ψ is a disjunction of atomic formulas.*

This fact can be straightforwardly proved by induction on the length of the formula.

To characterize the semantic types in $[\vee,\neg\neg]^n$ we will introduce a special type of IpL models, as we did previously for $[\wedge,\neg\neg]^n$.

3.4.5.45. DEFINITION. *A finite n-model K is called* n-minimal *if each node in K forces at most one atom and each terminal node forces at least one atom.*

3.4.5.46. THEOREM. *If ϕ and ψ are formulas in $[\vee,\neg\neg]^n$ such that $\phi \nvdash \psi$ then there is a node k in an n-minimal model such that $k \Vdash \phi$ and $k \nVdash \psi$.*

Proof. Let $\bigvee P \vee \bigvee\neg\neg\bigvee Q_i$ be the normal form of ϕ and $\bigvee R \vee \bigvee\neg\neg\bigvee S_j$ the normal form of ψ (where P, Q_i, R and the S_j are subsets of $\{p_1,\ldots,p_n\}$). As $\phi \nvdash \psi$ we have either some $p \in P$ which is not an element of R or some Q_i which is not a subset of any of the S_j.

In the first case a simple model of one node k such that $atom^n(k) = \{p\}$ is sufficient as a counter-example n-minimal model.

In the second case, let K be the n-minimal model with a root k_0 such that $atom^n(k_0) = \emptyset$ and terminal nodes k_q for every $q \in Q_i$. Then $k_0 \Vdash \neg\neg\bigvee Q_i$ but $k_0 \nVdash \bigvee R$ and also for every S_j, as Q_i is not a subset of S_j, we will have $k_0 \nVdash \neg\neg\bigvee S_j$. Hence $k_0 \Vdash \phi$ and $k_0 \nVdash \psi$ as required. $\dashv$

3.4.5.47. COROLLARY. *The fragment $[\vee,\neg\neg]^n$ is complete for n-minimal models.*

As we may confine our attention to n-minimal models in constructing an exact Kripke model for $[\vee,\neg\neg,\bot]^n$ we will use them in the definition of semantic types in $[\vee,\neg\neg]^n$.

3.4.5.48. DEFINITION. *Let k be a node in an n-minimal model. Then $\tau^n(k)$, the semantic type of k in $[\wedge,\vee,\neg]^n$ is the* semantic type *of k in $[\vee,\neg\neg]^n$ and $\phi^n(k)$, the* *type formula of k in $[\wedge,\vee,\neg]^n$ is the* type *of k in $[\vee,\neg\neg]^n$. The order of the semantic types in $[\vee,\neg\neg]^n$ is defined by:*

$$t \preceq t' \quad \Leftrightarrow \quad t = t' \textit{ or } t' \in j_1(t) \textit{ or } (j_0(t) \subseteq j_0(t') \textit{ and } \emptyset \neq j_1(t') \subseteq j_1(t)).$$

Note that for types in $[\vee,\neg\neg]^n$ we will have $\tau^n(k) \preceq \tau^n(l)$ if $atom^n(k) = atom^n(l)$ or $atom^n(k) = \emptyset$ and $\langle atom^n(l), \emptyset\rangle \in j_1(\tau^n(k))$.

In the definition of $\phi^n(k)$, the type of k in $[\vee,\neg\neg]^n$ one will recognize the normal form of the irreducible elements in the fragment.

Note that as k is a node in an n-minimal model $\phi^n(k)$ is either equivalent to an atomic formula or to $\neg\neg\bigvee Q$ where Q is the set of atoms forced in the terminal nodes above k. Hence modulo equivalence $\phi^n(k)$ is indeed a formula in $[\vee,\neg\neg]^n$.

Let in this subsection $Th^n(k)$ be the notation for the formulas in $[\vee,\neg\neg]^n$ forced by k.

3.4.5.49. Lemma. *If k and l are nodes in n-minimal models then:*

$$\tau^n(k) \preceq \tau^n(l) \quad \Leftrightarrow \quad Th^n(k) \subseteq Th^n(l) \quad \Leftrightarrow \quad l \Vdash \phi^n(k).$$

Proof. That $\tau^n(k) \preceq \tau^n(l)$ implies $Th^n(k) \subseteq Th^n(l)$ is a straightforward application of lemma 3.4.0.5, on n-minimal models. As $k \Vdash \phi^n(k)$, obviously, $Th^n(k) \subseteq Th^n(l)$ implies $l \Vdash \phi^n(k)$

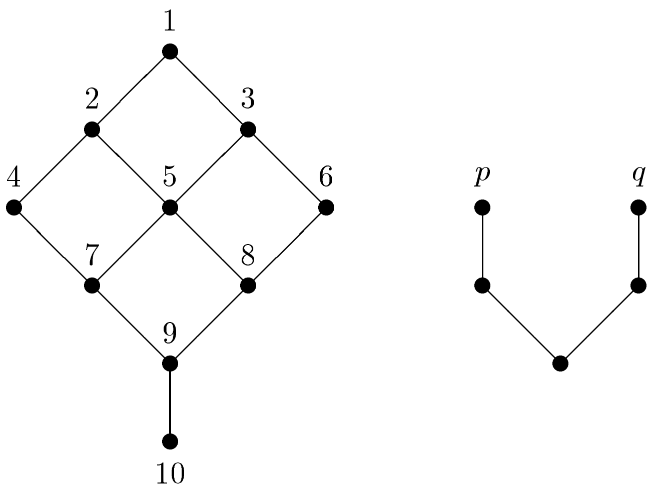
To prove that $l \Vdash \phi^n(k)$ implies $\tau^n(k) \preceq \tau^n(l)$, let $l \Vdash \phi^n(k)$. As a simple consequence we have $j_0(\tau^n(k)) \subseteq j_0(\tau^n(l))$. Hence, if $atom^n(k) = \{q\}$, then also $atom^n(l) = \{q\}$, as both k and l are nodes in n-minimal models. As a consequence, $atom^n(k) = \{q\}$ implies $\tau^n(k) = \tau^n(l)$. So, assume $atom^n(k) = \emptyset$, by the definition of an n-minimal model, k cannot be a terminal node. Observe, that $\phi^n(k)$ is a negation, equivalent to $\neg\neg\bigvee\{\phi^n(m) \mid m \in Ter(k)\}$.

If $atom^n(l) = \{q\}$, then, as l is a node in an n-minimal model, $\tau^n(l) = \langle\{q\}, \emptyset\rangle$ and from $l \Vdash \phi^n(k)$ we may conclude $l \Vdash \bigvee\{\phi^n(m) \mid m \in Ter(k)\}$ and hence $l \Vdash \phi^n(m)$, for some $m \in Ter(k)$. This proves that $\tau^n(l) \in j_1(\tau^n(k))$.

On the other hand, if $atom^n(l) = \emptyset$, then l cannot be (bisimular to) a terminal node in an n-minimal model. Hence $j_0(\tau^n(l)) \neq \emptyset$. To prove that $j_1(\tau^n(l)) \subseteq j_1(\tau^n(k))$, and hence $\tau^n(k) \preceq \tau^n(l)$, let $m \in Ter(l)$. Suppose $atom^n(m) = \{q\}$, then, as $m \Vdash \phi^n(k)$, $\phi^n(k) \nVdash \neg q$. From the definition of $\phi^n(k)$ infer that there is a $k' \in Ter(k)$ with $atom^n(k') = \{q\}$. Hence, $\tau^n(m) = \tau^n(k') \in j_1(\tau^n(k))$. So, we have $atom^n(k) = atom^n(l)$ and $\emptyset \neq j_1(\tau^n(l)) \subseteq j_1(\tau^n(k))$. Which, by definition, implies $\tau^n(k) \preceq \tau^n(l)$. ⊣

By ordering the semantic types in $[\vee,\neg\neg]^n$ we construct a n-minimal model $Exm([\vee,\neg\neg,\bot]^n)$ which consists of n terminal nodes, each forcing one of the atoms in $\{p_1,\ldots,p_n\}$ and non-terminal nodes that force no atomic formulas but are characterized by their set of terminal nodes.

Note that for $n > 1$ we added $\bot$ to the fragment to have a formula corresponding to the empty set in the exact model.



17. FIGURE. *The diagram of $[\vee, \neg\neg, \bot]^2$ and its exact Kripke model.*

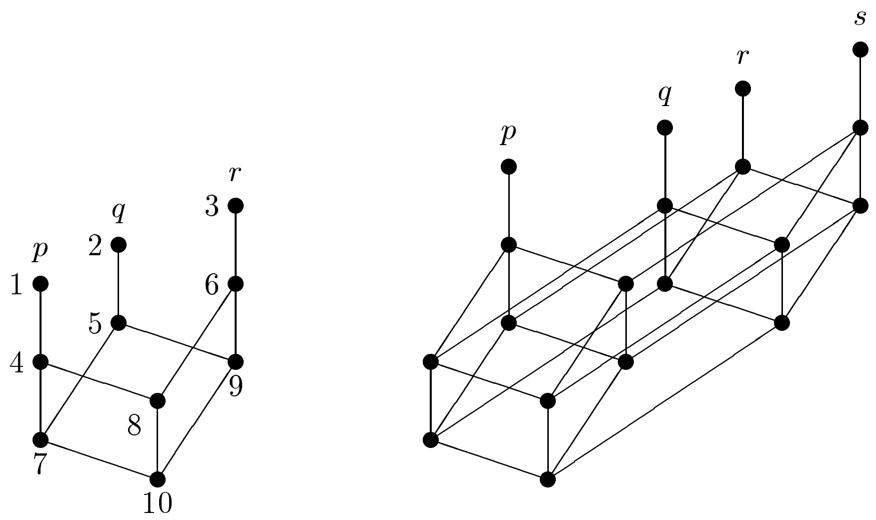
The formulas in the diagram of $[\vee, \neg\neg, \bot]^2$:

1. $\bot$	3. q	5. $p \vee q$	7. $\neg\neg p \vee q$	9. $\neg\neg p \vee \neg\neg q$
2. p	4. $\neg\neg p$	6. $\neg\neg q$	8. $p \vee \neg\neg q$	10. $\neg\neg(p \vee q)$

3.4.5.50. THEOREM. *The model $Exm([\vee, \neg\neg, \bot]^n)$ is the exact Kripke model of the fragment $[\vee, \neg\neg, \bot]^n$.*

Proof. Because $Exm([\vee, \neg\neg, \bot]^n)$ realizes all semantic types in $[\vee, \neg\neg]^n)$, the model is complete for the fragment. As we have seen, every semantic type corresponds to a type formula in $[\vee, \neg\neg, \bot]^n)$ and hence every non-empty closed subset corresponds exactly to a disjunction of these type formulas in $[\vee, \neg\neg, \bot]^n$. For the empty set the formula $\bot$ was added. $\dashv$

The construction of $Exm([\vee, \neg\neg, \bot]^n)$ shows that the non-terminal nodes correspond to non-empty subsets of $\{p_1, \ldots, p_n\}$ ordered by inclusion. Hence $Exm([\vee, \neg\neg, \bot]^n)$ is isomorphic to the the n-dimensional hypercube without a top, where the n maximal elements are connected to terminal nodes each forcing one of the atoms in $\{p_1, \ldots, p_n\}$.



18. FIGURE. *The exact Kripke models of* $[\vee, \neg\neg, \bot]^3$ *and* $[\vee, \neg\neg, \bot]^4$

As an example, we give the type formulas of $[\vee, \neg\neg, \bot]^3$.

1. p	5. $\neg\neg q$	9. $\neg\neg(q \vee r)$
2. q	6. $\neg\neg r$	10. $\neg\neg(p \vee q \vee r)$
3. r	7. $\neg\neg(p \vee q)$	
4. $\neg\neg p$	8. $\neg\neg(p \vee r)$	

Recall from subsection 3.3 that $D(k)$ is the k-th Dedekind number.

3.4.5.51. THEOREM.

$$|Diag([\vee, \neg\neg, \bot]^n)| = \sum_{k=0}^{n} \binom{n}{k} (D(k) + 1).$$

Proof. Observe that every formula in $[\vee, \neg\neg, \bot]^n$ is equivalent to a disjunction of atoms and formulas of the form $\neg\neg \bigvee R$ (where $\bigvee \emptyset = \bot$). Let $Q \subseteq \{p_1, \ldots, p_n\}$ and $|Q| = k$. It is not difficult to see that the set of formulas of the form $\neg\neg \bigvee R$, with $R \neq \emptyset$ and $R \subseteq \{p_1, \ldots, p_n\}$, ordered by $\vdash$, is isomorphic to $Diag([\vee]^n)$ (or equivalently $Diag([\wedge])$). Hence, the number of equivalence classes in $[\vee, \neg\neg, \bot]^n$ with a representative of the form $\bigvee Q \vee \neg\neg \bigvee R$ with $R \setminus Q \neq \emptyset$ equals $D(n-k)$.

As there are $\binom{n}{k}$ subsets $Q \subseteq \{p_1, \ldots, p_n\}$ with $|Q| = k$, we have, taking in account the cases where $R = \emptyset$:

$$|Diag([\vee, \neg\neg, \bot]^n)| = \sum_{k=0}^{n} \binom{n}{k} (D(n-k) + 1).$$

Now use $\binom{n}{k} = \binom{n}{n-k}$ to obtain the formula of the theorem. $\dashv$

3.5 The $[\wedge, \rightarrow, \neg]$ fragments

The diagrams of $[\wedge, \rightarrow, \neg]$ fragments have been studied by De Bruijn using exact models in [Bruijn 75a] as a special case of $[\wedge, \rightarrow]$ fragments[3]. Briefly stated the $[\wedge, \rightarrow, \neg]^n$ fragment is like a $[\wedge, \rightarrow]^{n+1}$ fragment, where one of the atoms is treated as $\perp$ (and hence $p_{n+1} \rightarrow \bigwedge\{p_1, \ldots, p_n\}$ will be true).

Using semantic types we may start with the $[\wedge, \rightarrow, \neg]$ fragments as the more 'natural' fragment.

Note that it is not trivial that the diagram of $[\wedge, \rightarrow, \neg]^n$ is a finite distributive lattice. Diego proved in [Diego 66] that the $[\wedge, \rightarrow]^n$ fragments are finite (and from the proof one could also infer that the diagram would be distributive). Using the above cited enbedding of $[\wedge, \rightarrow, \neg]^n$ into $[\wedge, \rightarrow]^{n+1}$ this implies that also $[\wedge, \rightarrow, \neg]^n$ will have a finite diagram.

As for the lattice operations in the diagram of $[\wedge, \rightarrow, \neg]^n$, it will be obvious how $\wedge$ will act as $\cap$, but for $\cup$ there is no simple operation in $[\wedge, \rightarrow, \neg]^n$ as $\vee$ is not definable in terms of $\{\wedge, \rightarrow, \neg\}$. Hence in the following subsections the reader should be cautious in not taking the irreducible formulas in (subfragments of) $[\wedge, \rightarrow, \neg]^n$ as $\vee$-irreducible formulas.

To define the semantic types in $[\wedge, \rightarrow, \neg]$ fragments we will restrict our models to the *$\cap$-independent models.*

3.5.0.1. DEFINITION. *Let K be a finite* **IpL***-model model and $k \in K$. k is* $\cap$-independent *if k is a* terminal node *or:*

$$atom(k) \neq \bigcap\{atom(l) \mid k < l\}.$$

A finite **IpL***-model K is* $\cap$-independent *if every node $k \in K$ is $\cap$-independent.*

Observe that, in a $\cap$-independent n-model, $k < l$ implies $atom^n(k) \neq atom^n(l)$.

As is not difficult to see, $\cap$-independentness is preserved under taking submodels.

3.5.0.2. DEFINITION. *Let K be a finite* **IpL***-model. The* $\cap$-independent reduction *of K is the model $K^\cap$, with the $\cap$-independent nodes of K as its worlds and its accessibility relation inherited from K.*

Note that, as $\cap$-independentness is preserved by taking submodels, the $\cap$-independent reduction is an $\cap$-independent model.

3.5.0.3. THEOREM. *Let K be a finite* **IpL***-model and $k \in K$, then for all formulas $\phi \in [\wedge, \rightarrow, \neg]$:*

$$k \Vdash \phi \quad \Leftrightarrow \quad (\uparrow k)^\cap \Vdash \phi.$$

Proof. By induction on $\delta(k)$. If $\delta(k) = 0$, then the theorem is trivial. So assume $\delta(k) = m + 1$ and apply induction on the length of ϕ. The only interesting case is $\phi = \psi \rightarrow \chi$ (including negation as a special case).

[3] This was also the approach in [Hendriks 93]

For the proof in the $\Rightarrow$-direction, let $k \Vdash \psi \to \chi$. To prove $(\uparrow k)^\cap \Vdash \psi \to \chi$, let $l \in (\uparrow k)^\cap$ and $(\uparrow l)^\cap \Vdash \psi$. By the induction hypothesis, $l \Vdash \psi$. As $k \leq l$, we conclude $l \Vdash \chi$ and, again by the induction hypothesis, $(\uparrow l)^\cap \Vdash \chi$. Which proves $(\uparrow k)^\cap \Vdash \psi \to \chi$

For the proof in the $\Leftarrow$-direction, let $(\uparrow k)^\cap \Vdash \psi \to \chi$. To prove $k \Vdash \psi \to \chi$, let $k \leq l$ and $l \Vdash \psi$. By the induction hypothesis, $(\uparrow l)^\cap \Vdash \psi$ and as obviously $(\uparrow l)^\cap \subseteq (\uparrow k)^\cap$, $(\uparrow l)^\cap \Vdash \chi$ and, again by the induction hypothesis, $l \Vdash \chi$. Which proves $k \Vdash \psi \to \chi$. $\dashv$

The converse of theorem 3.5.0.3 also holds, as theorem 3.5.0.24 below proves.

3.5.0.4. THEOREM. *The $[\wedge, \to, \neg]$ fragment is complete for $\cap$-independent* **IpL** *models.*

Proof. Obvious using theorem 3.5.0.3. $\dashv$

Now we are ready to define the semantic types for $[\wedge, \to, \neg]^n$ of nodes in $\cap$-independent **IpL**-models.

3.5.0.5. DEFINITION. *For k a node in a finite $\cap$-independent n-model, we define $\tau^n(k)$, the* semantic type *of k in $[\wedge, \to, \neg]^n$ as:*

$$\tau^n(k) = \langle atom^n(k), \{\tau^n(l) \mid k < l\}\rangle.$$

If t and t' are semantic types in $[\wedge, \to, \neg]^n$ then define $t \preceq t'$ if $t = t'$ or $t' \in j_1(t)$.

The definition is sound, as in $\cap$-independent n-models $k < l$ implies $atom^n(k) \neq atom^n(l)$ and hence, it is excluded that $\tau^n(k) \in j_1(\tau^n(k)$.

The following simple lemma proves that semantic types in $[\wedge, \to, \neg]^n$ indeed behave as expected.

3.5.0.6. LEMMA. *For nodes k and l in finite $\cap$-independent n-models, define $k \sim l$ if $\tau^n(k) = \tau^n(l)$. Then $\sim$ is a bisimulation.*

Proof. That $k \sim l$ implies $atom^n(k) = atom^n(l)$ is trivial. As the other conditions in definition 2.2.0.1 are symmetric, we only prove one of them. Let $k < k'$ then, by definition, $\tau^n(k') \in j_1(\tau^n(k))$. From $j_1(\tau^n(k)) = j_1(\tau^n(l))$ infer that there is a $l' > l$ such that $\tau^n(k') = \tau^n(l')$. $\dashv$

3.5.0.7. COROLLARY. *If k and l are nodes in finite $\cap$-independent n-models then $\tau^n(k) \preceq \tau^n(l)$ implies $Th^n(k) \subseteq Th^n(l)$.*

Proof. If $\tau^n(k) = \tau^n(l)$ then we have $k \rightleftarrows^n l$ and hence $Th^n(k) = Th^n(l)$. Otherwise, from $\tau^n(l) \in j_1(\tau^n(k))$ infer that there is a $k' > k$ such that $\tau^n(k') = \tau^n(l)$ and hence $Th^n(k) \subseteq Th^n(k') = Th^n(l)$. $\dashv$

We are almost ready now to introduce the exact Kripke model of $[\wedge, \to, \neg]^n$ as the ordered set of semantic types in $[\wedge, \to, \neg]^n$.

First however we have to prove that there are only finitely many semantic types in $[\wedge, \to, \neg]^n$. To do so we use the following lemma.

3.5.0.8. LEMMA. *If t is a semantic type in $[\wedge, \rightarrow, \neg]^n$, then there is a node k in a finite $\cap$-independent n-model such that $\tau^n(k) = t$ and $\delta(k) \leq n - |j_0(t)|$.*

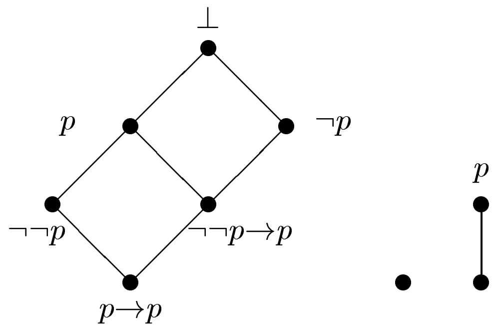
Proof. By a simple induction on $d = n - |j_0(t)|$. For $d = 0$ observe that $j_0(t) = \{p_1, \ldots, p_n\}$. As t has to be a semantic type of a node in an $\cap$-independent n-model, we may infer that $j_1(t) = \emptyset$.

If $d > 0$ then by induction hypothesis every $t' \in j_1(t)$ is realizable by a node $k_{t'}$ with depth at most $d - 1$ (again using the fact that t must be a type of a node in an $\cap$-independent n-model). Of course if a node k with $atom^n(k) = j_0(t)$ is put below all of these $k_{t'}$ (with $t' \in j_1(t)$), then $\tau^n(k) = t$ and $\delta(k) \leq d$. $\dashv$

3.5.0.9. COROLLARY. *There are finitely many semantic types in $[\wedge, \rightarrow, \neg]^n$.*

Proof. Note that there are only finitely many $\cap$-independent n-models with depth less or equal than n and every semantic type in $[\wedge, \rightarrow, \neg]^n$ can be realized in one of these models. $\dashv$

3.5.0.10. DEFINITION. *If T is the set of semantic types in $[\wedge, \rightarrow, \neg]^n$ then define $Exm([\wedge, \rightarrow, \neg]^n) = \langle T, \preceq, j_0 \rangle$.*



19. FIGURE. *The diagram of $[\wedge, \rightarrow, \neg]^1$ and the model $Exm([\wedge, \rightarrow, \neg]^1)$.*

Note that we somewhat prematurely baptized the model defined as an exact model, but we will prove this claim in due course. First there are some simple facts to be arrested. If ϕ a formula in $[\wedge, \rightarrow, \neg]^n$, then $[\![\phi]\!]$ will be the valuation of ϕ in $Exm([\wedge, \rightarrow, \neg]^n)$ (hence $[\![\phi]\!] = \{k \in Exm([\wedge, \rightarrow, \neg]^n) \mid k \Vdash \phi\}$).

3.5.0.11. FACTS.

1. *$Exm([\wedge, \rightarrow, \neg]^n)$ is a finite $\cap$-independent n-model;*
2. *if t a semantic type in $[\wedge, \rightarrow, \neg]^n$ then $\tau^n(t) = t$ in $Exm([\wedge, \rightarrow, \neg]^n)$;*
3. *$Exm([\wedge, \rightarrow, \neg]^n)$ is complete for $[\wedge, \rightarrow, \neg]^n$: if ϕ and ψ in $[\wedge, \rightarrow, \neg]^n$ we have*

$$\phi \vdash \psi \quad \Leftrightarrow \quad [\![\phi]\!] \subseteq [\![\psi]\!].$$

The first two of these facts are established by a close inspection of the construction of $Exm([\wedge, \rightarrow, \neg]^n)$ from the semantic types in $[\wedge, \rightarrow, \neg]^n$. The last one is a simple consequence of the lemmas above.

To prove $Exm([\wedge, \rightarrow, \neg]^n)$ to be the exact model of $[\wedge, \rightarrow, \neg]^n$, we need a formula in $[\wedge, \rightarrow, \neg]^n$ for every closed subset of $Exm([\wedge, \rightarrow, \neg]^n)$. Unfortunately there is no known simple construction for such a formula, which is independent of the construction of the exact model. However, there is an easy way out.

3.5.0.12. LEMMA. *The diagram of* $[\wedge, \rightarrow, \neg]^n$ *is finite.*

Proof. Observe that the equivalence class of a $[\wedge, \rightarrow, \neg]^n$ formula ϕ corresponds to $[\![\phi]\!]$ in $Exm([\wedge, \rightarrow, \neg]^n)$. That is $\phi \equiv \psi$ iff $[\![\phi]\!] = [\![\psi]\!]$. As $Exm([\wedge, \rightarrow, \neg]^n)$ is finite, there are only finitely many equivalence classes in $[\wedge, \rightarrow, \neg]^n$. ⊣

De Bruijn proved the finiteness of $Diag([\wedge, \rightarrow, \neg]^n)$ in [Bruijn 75a]. Diego and Urquhart independently proved that $Diag([\rightarrow]^n)$ is finite (see [Diego 66] and [Urquhart 74]), from which the finiteness of $Diag([\wedge, \rightarrow, \neg]^n)$ is a simple corollary. Observe, that using $p \wedge q \rightarrow r \wedge s \equiv (p \rightarrow (q \rightarrow r)) \wedge (p \rightarrow (q \rightarrow s))$, we can prove that every formula in $[\wedge, \rightarrow, \neg]^n$ is a conjunction of formulas in $[\rightarrow, \neg]^n$. As obviously $|Diag([\rightarrow, \neg]^n)| \leq |Diag([\rightarrow]^{n+1})|$, the finiteness of $Diag([\rightarrow]^n)$ (for each n) implies that $Diag([\wedge, \rightarrow, \neg]^n)$ is finite.

3.5.0.13. COROLLARY. *For every node k in a finite $\cap$-independent n-model there are, up to equivalence, only finitely many formulas of* $[\wedge, \rightarrow, \neg]^n$ *in* $Th^n(k)$.

The corollary justifies the following definitions.

3.5.0.14. DEFINITION. *For a node k in a finite $\cap$-independent n-model define,* $\phi^n(k)$, *the* type *of k in* $[\wedge, \rightarrow, \neg]^n$ *as*

$$\phi^n(k) = \bigwedge Th^n(k).$$

As stated earlier, this is an easy way out and we will return to the construction of type formulas in the sequel. Obviously $\phi^n(k)$ is an axiom for $Th^n(k)$ and $\tau^n(k) \preceq \tau^n(l)$ then $l \Vdash \phi^n(k)$.

3.5.0.15. LEMMA. *Let k and l be nodes in finite $\cap$-independent n-models. If $l \Vdash \phi^n(k)$ then $\tau^n(k) \preceq \tau^n(l)$.*

Proof. Assume $l \Vdash \phi^n(k)$. To prove $\tau^n(k) \preceq \tau^n(l)$ we will use induction on $\delta(l)$, the depth of l. If $\delta(l) = 0$ then $k \nVdash \neg\phi^n_{\mathbf{CpL}}(l)$ and hence there is a $k' > k$ such that $k' \Vdash \phi^n_{\mathbf{CpL}}(l)$. In a $\cap$-independent n-model we may infer that k' has to be a terminal node and hence $\tau^n(k') = \tau^n(l)$. Which proves $\tau^n(k) \preceq \tau^n(l)$.

If $\delta(l) > 0$, let $l' > l$. Then $l' \Vdash \phi^n(k)$ and $\delta(l') < \delta(l)$. According to the induction hypothesis we will have $\tau^n(k) \preceq \tau^n(l')$. This proves that $j_1(\tau^n(l)) \subseteq j_1(\tau^n(k))$. As obviously the assumption implies that $atom^n(k) \subseteq atom^n(l)$, this proves $\tau^n(k) \preceq \tau^n(l)$. ⊣

3.5.0.16. COROLLARY. *If k and l are nodes in finite $\cap$-independent n-models then:*

$$\tau^n(k) \preceq \tau^n(l) \quad \Leftrightarrow \quad Th^n(k) \subseteq Th^n(l) \quad \Leftrightarrow \quad l \Vdash \phi^n(k).$$

Proof. Corollary 3.5.0.7 takes care of $\tau^n(k) \preceq \tau^n(l) \Rightarrow Th^n(k) \subseteq Th^n(l)$. As $\phi^n(k)$ is the axiom of $Th^n(k)$, obviously $Th^n(k) \subseteq Th^n(l)$ implies $l \Vdash \phi^n(k)$. Finally, $l \Vdash \phi^n(k) \Rightarrow \tau^n(k) \preceq \tau^n(l)$ by lemma 3.5.0.15. $\dashv$

With corollary 3.5.0.16 we are ready to prove that $Exm([\wedge, \rightarrow, \neg]^n)$ is indeed the exact Kripke model we were looking for.

3.5.0.17. THEOREM. *The model $Exm([\wedge, \rightarrow, \neg]^n)$ defined above is the exact Kripke model of $[\wedge, \rightarrow, \neg]^n$.*

Proof. As noted before, $Exm([\wedge, \rightarrow, \neg]^n)$ is complete for $[\wedge, \rightarrow, \neg]^n$ and we have to prove that every closed subset X in this model corresponds to a formula in $[\wedge, \rightarrow, \neg]^n$.

To do so we apply essentially use the same trick that was used to define the types of nodes in $[\wedge, \rightarrow, \neg]^n$. Let $\phi^n(X) = \bigwedge \bigcap \{Th^n(k) \mid k \in X\}$. Then clearly, by definition, it will be true that $X \subseteq [\![\phi^n(X)]\!]$.

To prove the inclusion in the other direction, suppose that $k \Vdash \phi^n(X)$. With induction on $\delta(k)$, the depth of k, we will prove that $k \in X$. If $\delta(k) = 0$ then k is a terminal node and apparently it is the case that for some $l \in X$ we had $l \nVdash \neg\phi^n_{\mathbf{CpL}}(k)$. As otherwise $\phi^n(X)$ would imply $\neg\phi^n_{\mathbf{CpL}}(k)$. Hence for some $l \in X$ there is a $l' > l$ with $l' \Vdash \phi^n_{\mathbf{CpL}}(k)$. As $Exm([\wedge, \rightarrow, \neg]^n)$ is a $\cap$-independent n-model, this l' has to be a terminal node. As the semantic types in $Exm([\wedge, \rightarrow, \neg]^n)$ are unique, we conclude that $k = l'$. From $l \in X$ and $l < k$ infer that $k \in X$ as X is a closed subset of $Exm([\wedge, \rightarrow, \neg]^n)$.

If $\delta(k) > 0$ then for $k' > k$ we conclude from $k' \Vdash \phi^n(X)$ and the induction hypothesis that $k' \in X$. As $Exm([\wedge, \rightarrow, \neg]^n)$ is a $\cap$-independent n-model, there is a $q \in atom^n(k) \setminus \bigcap\{atom^n(l) \mid k < l\}$. Note that for $k < l$ we have $l \Vdash \phi^n(k) \rightarrow q$ but $k \nVdash \phi^n(k) \rightarrow q$. Now suppose that $l \in X$ and $l \Vdash \phi^n(k)$ then by lemma 3.5.0.15 we have $k \leq l$. Hence $(\phi^n(k) \rightarrow q) \in X$ iff $k \notin X$. As $k \Vdash \phi^n(X)$ infer that $k \in X$. $\dashv$

The model $Exm([\wedge, \rightarrow, \neg]^n)$ can stagewise be constructed as the minimal $\cap$-independent n-model realizing all semantic types in $[\wedge, \rightarrow, \neg]^n$. Let us define the $n+1$ stages E^n_i needed in the construction. Recall that $\mathcal{P}^*(X)$ is the set of closed subsets in X.

3.5.0.18. DEFINITION. *Define E^n_0 as the set of 2^n terminal nodes with semantic type $\langle Q, \emptyset \rangle$ such that $Q \subseteq \{p_1, \ldots, p_n\}$.*

Now inductively define:

$$E^n_{m+1} = E^n_m \cup \{\langle Q, S \rangle \mid S \in \mathcal{P}^*(E^n_m) \text{ and } Q \subset \bigcap\{j_0(t) \mid t \in S\} \neq Q\}.$$

The order in E^n_m is the order of types in $[\wedge, \rightarrow, \neg]^n$.

Note that the construction of E^n_m is only possible for $m \leq n$.

3.5.0.19. FACTS. *From the construction of E_n^n the following facts are obvious:*

1. *E_n^n is a finite $\cap$-independent n-model;*
2. *Every semantic type of $[\wedge,\rightarrow,\neg]^n$ is realized in E_n^n exactly once;*
3. *$E_n^n = Exm([\wedge,\rightarrow,\neg]^n)$.*

Let us return to the type formulas in $[\wedge,\rightarrow,\neg]^n$. Recall the definition of $\phi^n(k)$ in definition 3.5.0.14.

3.5.0.20. DEFINITION. *Let k be a node in a finite $\cap$-independent n-model and $X \subseteq \{p_1,\ldots,p_n\}$. Define:*

1. *$Newatom^n(k) = \{q \mid q \in \bigcap\{atom^n(l) \mid k < l\} \setminus atom^n(k)\}$;*
2. *$\Delta X = \bigwedge\{p \rightarrow q \mid p, q \in X\}$;*
3. $\psi^n(k) = \begin{cases} \neg\phi^n_{\mathbf{CpL}}(k) & \textit{if } \delta(k) = 0 \\ \\ \phi^n(k) \rightarrow q, \textit{ where } q \in Newatom^n(k) & \textit{otherwise.} \end{cases}$

The proper definition of $\psi^n(k)$ of course requires a choice of $q \in Newatom^n(k)$. As this choice will not make any difference in the sequel, one may take for example the p_i with the least i such that $p_i \in Newatom^n(k)$.

3.5.0.21. LEMMA. *If k and l are nodes in finite $\cap$-independent n-models then:*

$$l \nVdash \psi^n(k) \quad \Leftrightarrow \quad \tau^n(l) \preceq \tau^n(k).$$

Proof. If k is a terminal node, the lemma is rather trivial. So, assume $\delta(k) > 0$. To prove $l \nVdash \psi^n(k) \;\Rightarrow\; \tau^n(l) \preceq \tau^n(k)$, let $l \nVdash \psi^n(k)$. As $\psi^n(k) = \phi^n(k) \rightarrow q$, this implies, for some $l' \geq l$, that $l' \Vdash \phi^n(k)$ and $l' \nVdash q$, where $q \in Newatom^n(k)$. According to corollary 3.5.0.16, $l' \Vdash \phi^n(k)$ implies $\tau^n(k) \preceq \tau^n(l')$. In finite $\cap$-independent models, it is not difficult to prove that if $\tau^n(k) \prec \tau^n(m)$ (i.e. $\tau^n(k) \preceq \tau^n(m)$ but $\tau^n(k) \neq \tau^n(m)$), then $m \Vdash q$, for $q \in Newatom^n(k)$. As $l' \nVdash q$ and obviously from $l \leq t'$ we may conclude that $\tau^n(l) \preceq \tau^n(l')$, we have $\tau^n(l) \preceq \tau^n(l') = \tau^n(k)$.

To prove $\tau^n(l) \preceq \tau^n(k) \;\Rightarrow\; l \nVdash \psi^n(k)$, observe that by definition $k \nVdash q$. So, if $\tau^n(l) \preceq \tau^n(k)$, then $l \Vdash \psi^n(k)$ would imply, by corollary 3.5.0.16, that $k \Vdash \psi^n(k)$. As $k \Vdash \phi^n(k)$, we would have $k \Vdash q$, a contradiction. Hence, we conclude $l \nVdash \psi^n(k)$. $\dashv$

We are now ready for a characterization of $\phi^n(k)$, the type of k in $[\wedge,\rightarrow,\neg]^n$. An analogous characterization was used, as a definition, in [De Jongh 68] (also in [De Jongh 70], [De Jongh 80] and [JHR 91]). We will use the exact model of $[\wedge,\rightarrow,\neg]^n$ in the characterization. Note that as for every semantic type in a finite $\cap$-independent n-model, there is a node in the exact model with the same semantic type, theorem 3.5.0.23 is more generally applicable.

3.5.0.22. DEFINITION. *If k is a node in $Exm([\wedge, \rightarrow, \neg]^n)$ and $q \in Newatom^n(k)$ then define:*

$$\Phi^n(k) = \begin{cases} \phi^n_{\mathbf{CpL}}(k) & \text{if } \delta(k) = 0 \\ & \\ \bigwedge atom^n(k) \wedge \Delta Newatom^n(k) \wedge & \\ \quad \bigwedge\{\psi^n(l) \rightarrow q \mid k <_1 l\} \wedge & \\ \quad \bigwedge\{\psi^n(m) \mid not\,(m \leq k) \text{ and} & \\ \qquad \bigcap\{atom^n(l) \mid k < l\} \subseteq atom^n(m)\} & \text{if } \delta(k) > 0. \end{cases}$$

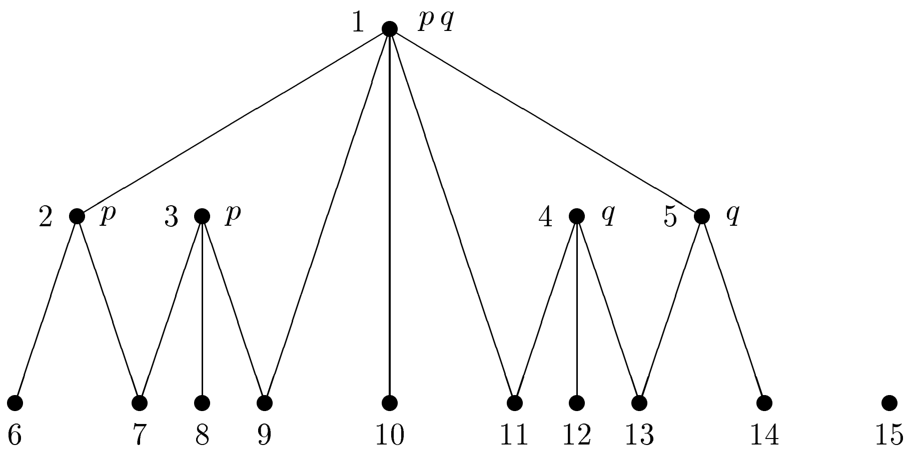
3.5.0.23. THEOREM. *If k is a node in $Exm([\wedge, \rightarrow, \neg]^n)$ then $\phi^n(k) \equiv \Phi^n(k)$.*

Proof. If k is a terminal node, then it is rather obvious that $\phi^n(k) = \phi^n_{\mathbf{CpL}}(k)$ and hence the theorem is true by definition. So assume $\delta(k) > 0$.

To prove $\phi^n(k) \vdash \Phi^n(k)$ we show that $k \Vdash \Phi^n(k)$. That $k \Vdash \bigwedge atom^n(k) \wedge \Delta Newatom^n(k) \wedge$ is rather obvious. For $k < l$ we have $l \Vdash q$ and $k \nVdash \psi^n(l)$ by lemma 3.5.0.21. According to the same lemma $k \Vdash \psi^n(m)$ if not $m \leq k$, which proves that k will also force the last of the conjunctions in $\Phi^n(k)$.

For the proof of the other direction, assume $l \Vdash \Phi^n(k)$. We will show that as a consequence $k \leq l$ and hence $l \Vdash \phi^n(k)$. As $Exm([\wedge, \rightarrow, \neg]^n)$ is the exact model of $[\wedge, \rightarrow, \neg]^m$, this proves $\Phi^n(k) \vdash \phi^n(k)$.

Suppose $Newatom^n(k) \subseteq atom^n(l)$. Then, using the last part in the conjunction of $\Phi^n(k)$, not $k \leq l$ implies $\Phi^n(k) \vdash \psi^n(l)$. As $l \nVdash \psi^n(l)$, infer that $k \leq l$ and hence $l \Vdash \phi^n(k)$. If $Newatom^n(k)$ is not a subset of $atom^n(l)$, then $l \nVdash q$ for every $q \in Newatom^n(k)$ (because $l \Vdash \Delta Newatom^n(k)$). Hence if $k <_1 k'$ then, using the third conjunct in $\Phi^n(k)$, we have $l \nVdash \psi^n(k')$. By lemma 3.5.0.21 this implies that $l \leq k'$. Hence $atom^n(l)$ will be included in $atom^n(k) \cup Newatom^n(k)$. From $atom^n(k) \subseteq atom^n(k')$ and $Newatom^n(k) \cap atom^n(l) = \emptyset$ infer that $atom^n(l) = atom^n(k)$ and hence $\tau^n(k) = \tau^n(l)$. As semantic types are unique in $Exm([\wedge, \rightarrow, \neg]^n)$, we conclude $k = l$ and trivially $l \Vdash \phi^n(k)$. $\dashv$

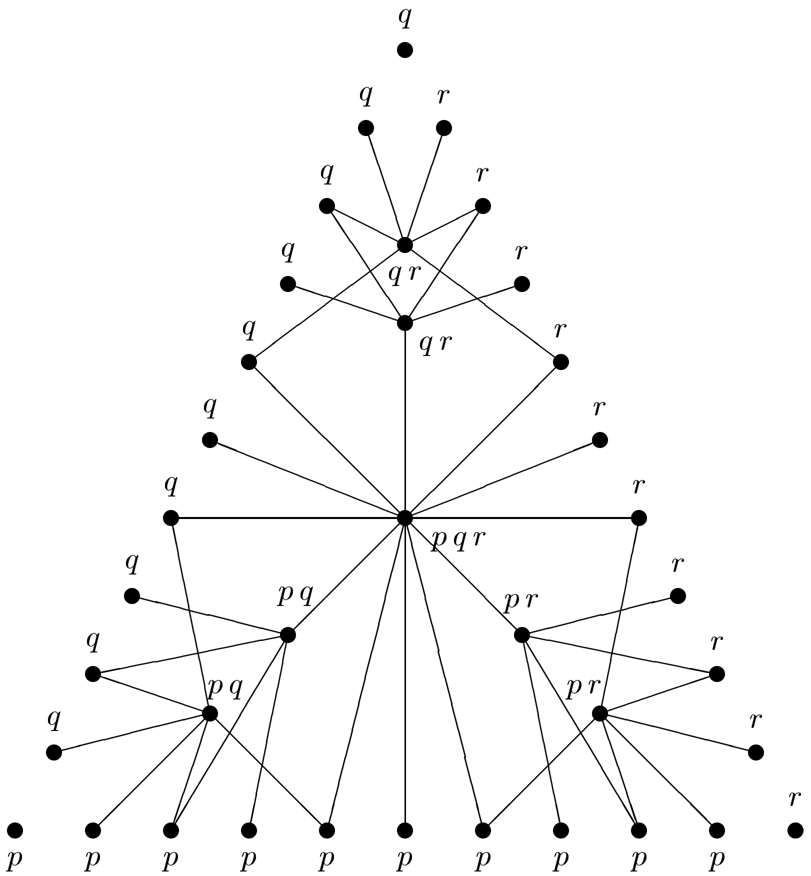


20. FIGURE. *The exact Kripke model of $[\wedge, \rightarrow, \neg]^2$.*

This model has 2 134 upwards closed subsets, corresponding to the 2 134 equivalence classes of $[\wedge, \rightarrow, \neg]^2$.

The type formulas in $[\wedge, \rightarrow, \neg]^2$ are:

1. $p \wedge q$
2. $p \wedge \neg\neg q$
3. $p \wedge \neg q$
4. $q \wedge \neg p$
5. $q \wedge \neg\neg p$
6. $\neg\neg q \wedge ((p \rightarrow q) \rightarrow p)$
7. $(\neg\neg q \rightarrow p) \wedge ((\neg\neg q \rightarrow q) \rightarrow p)$
8. $\neg(p \rightarrow q)$
9. $(q \rightarrow p) \wedge (\neg q \rightarrow p) \wedge (\neg\neg q \rightarrow q)$
10. $(p \leftrightarrow q) \wedge \neg\neg p$
11. $(p \rightarrow q) \wedge (\neg p \rightarrow q) \wedge (\neg\neg p \rightarrow p)$
12. $\neg(q \rightarrow p)$
13. $(\neg\neg p \rightarrow q) \wedge ((\neg\neg p \rightarrow p) \rightarrow q)$
14. $\neg\neg p \wedge ((q \rightarrow p) \rightarrow q)$
15. $\neg p \wedge \neg q$



21. FIGURE. *Part of the model $Exm([\wedge, \rightarrow, \neg]^3)$. The 6 386 nodes k with $atom^n(k) = \emptyset$ have been omitted. The order in the model is from the outside inwards.*

We may now use the exact model of the fragment $[\wedge, \rightarrow, \neg]^n$ to prove the converse of theorem 3.5.0.3. This was suggested first by Albert Visser.

3.5.0.24. THEOREM. *If ϕ is an* **IpL** *formula such that for every node k in a finite Kripke model:*

$$k \Vdash \phi \quad \Leftrightarrow \quad (\uparrow k)^{\cap} \Vdash \phi$$

then ϕ is equivalent to a formula in $[\wedge, \rightarrow, \neg]^n$.

Proof. Let ϕ be a formula in $\mathbf{IpL}^n$ with the property that for every Kripke model K and every node $k \in K$, $k \Vdash \phi \quad \Leftrightarrow \quad (\uparrow k)^\cap \Vdash \phi$. Let $\chi \in [\wedge, \rightarrow, \neg]^n$ be the formula with $[\![\chi]\!] = [\![\phi]\!]$ in $Exm([\wedge, \rightarrow, \neg]^n)$. For a node k in a finite $\cap$-independent model we have, using lemma 3.5.0.6: $k \Vdash \phi \quad \Leftrightarrow \quad k \Vdash \chi$.

As χ is a formula in $[\wedge, \rightarrow, \neg]^n$, by theorem 3.5.0.3, $k \Vdash \chi \quad \Leftrightarrow \quad (\uparrow k)^\cap \Vdash \chi$. Hence we have:

$$k \Vdash \phi \quad \Leftrightarrow \quad (\uparrow k)^\cap \Vdash \phi \quad \Leftrightarrow \quad (\uparrow k)^\cap \Vdash \chi \quad \Leftrightarrow \quad k \Vdash \chi.$$

Which proves $\phi \equiv \chi$. $\dashv$

3.5.1 The $[\rightarrow, \neg]$ fragments

To calculate the diagram of $[\rightarrow, \neg]^n$ we have to use the exact Kripke model of $[\wedge, \rightarrow, \neg]^n$, as $Diag([\rightarrow, \neg]^n)$ for $n > 1$ is not a lattice and hence does not have an exact model of its own.

3.5.1.25. LEMMA. *Every formula in $[\wedge, \rightarrow, \neg]^n$ is equivalent to a conjunction of formulas in $[\rightarrow, \neg]^n$*

Proof. We proceed by induction on the length of ϕ. Only the cases in which ϕ is a negation or an implication are non-trivial. If $\phi = \neg\psi$, then according to the induction hypothesis ψ is a conjunction of formulas in $[\rightarrow, \neg]^n$. Now apply the **IpL** theorem $\neg(A \wedge B) \equiv A \rightarrow \neg B$ to show that ϕ is equivalent to a formula in $[\rightarrow, \neg]^n$.

In the case that $\phi = \psi \rightarrow \chi$, we use the induction hypothesis first to infer that ϕ is equivalent to a conjunction of formulas of the form $\psi \rightarrow \upsilon_i$, where $\chi = \bigwedge \upsilon_i$ and every υ_i is a formula in $[\rightarrow, \neg]^n$. Again applying both the induction hypothesis and the theorem $A \wedge B \rightarrow C \equiv A \rightarrow (b \rightarrow C)$, we conclude that ϕ is equivalent to a conjunction of formulas in $[\rightarrow, \neg]^n$. $\dashv$

3.5.1.26. LEMMA. *An **IpL** formula ϕ is equivalent to a formula in $[\rightarrow, \neg]^n$ iff $\phi \equiv \neg\psi$ or $\phi \equiv \psi \rightarrow p$, for some $\psi \in [\wedge, \rightarrow, \neg]^n$ and $p \in \{p_1, \ldots, p_n\}$.*

Proof. That every formula $\phi \in [\rightarrow, \neg]^n$ is equivalent to either a negation or a formula $\psi \rightarrow p$ with $\psi \in [\wedge, \rightarrow, \neg]^n$ and $p \in \{p_1, \ldots, p_n\}$ can easily be proved by induction on the length of ϕ. If $\phi = \psi \rightarrow \chi$, note that by the induction hypothesis $\chi \equiv \upsilon \rightarrow p$ and hence $\phi \equiv \psi \wedge \upsilon \rightarrow p$.

For the other direction of the lemma, note that if $\phi \equiv \psi \rightarrow p$ with $\phi \in [\wedge, \rightarrow, \neg]^n$, then ψ is, according to lemma 3.5.1.25 equivalent to a conjunction of formulas in $[\rightarrow, \neg]^n$. Now apply $A \wedge B \rightarrow C \equiv A \rightarrow (B \rightarrow C)$. $\dashv$

For the calculation of the number of classes in $Diag([\rightarrow, \neg]^n)$, it is more convenient to work with the complement of $[\![\phi]\!]$ in $Exm([\wedge, \rightarrow, \neg]^n)$.

3.5.1.27. Definition. *Let $[\![\phi]\!]$ be the valuation of formulas in $Exm([\wedge, \rightarrow, \neg]^n)$. Define $\alpha^n(\phi) = Exm([\wedge, \rightarrow, \neg]^n) \setminus [\![\phi]\!]$.*

3.5.1.28. Lemma. *Let ϕ and ψ be formulas in $[\wedge, \rightarrow, \neg]^n$. Then*

1. $\alpha^n(\phi) \subseteq \alpha^n(\psi) \quad \Leftrightarrow \quad \psi \vdash \phi$;
2. $\alpha^n(\phi \wedge \psi) = \alpha^n(\phi) \cup \alpha^n(\psi)$;
3. $\alpha^n(\phi \rightarrow \psi) = \downarrow(\alpha^n(\psi) \setminus \alpha^n(\phi))$;
4. $\alpha^n(\neg\phi) = \downarrow(Exm([\wedge, \rightarrow, \neg]^n) \setminus \alpha^n\phi) = \downarrow[\![\phi]\!]$.

Proof. The proofs of the first two propositions in the lemma are straightforward. The last part of the lemma is a simple corollary of the third.

For proof of the third statement in the lemma, observe that by the definition of α^n: $k \in \alpha^n(\phi \rightarrow \psi)$ iff for some $l \geq k$ both $l \Vdash \phi$ and $l \nVdash \psi$. Hence $k \in \alpha^n(\phi \rightarrow \psi)$ iff for some $l \geq k$ we have $l \in \alpha^n(\phi) \setminus \alpha^n(\psi)$. But the latter is equivalent to $k \in \downarrow(\alpha^n(\psi) \setminus \alpha^n(\phi))$. $\dashv$

3.5.1.29. Definition. *For a formula ϕ in $[\wedge, \rightarrow, \neg]^n$ we define $ucv^n(\phi)$,* the upper carrier *of ϕ, as the set of maximal elements in $\alpha^n(\phi)$.*

The upper carrier valuation was introduced in [Bruijn 75a]. Using the dual of our exact models, De Bruijn, called it the *lower carrier valuation.* Observe that $\alpha^n(\phi) = \downarrow ucv^n(\phi)$ and $ucv^n(\phi)$ is the smallest subset in $Exm([\wedge, \rightarrow, \neg]^n)$ with this property.

3.5.1.30. Lemma. *For $\phi \in [\wedge, \rightarrow, \neg]^n$ let $An^n(\phi)$ be the set of equivalence classes in $[\wedge, \rightarrow, \neg]^n$ that have a representative of the form $\psi \rightarrow \phi$, with $\psi \in [\wedge, \rightarrow, \neg]^n$. Then*

$$|An^n(\phi)| = |\mathcal{P}(ucv^n(\phi))| = 2^{|ucv^n(\phi)|}.$$

Proof. As $\alpha^n(\psi \rightarrow \phi) = \downarrow(\alpha^n(\psi) \setminus \alpha^n(\phi)) = \downarrow(ucv^n(\phi) \setminus \alpha^n(\psi))$, every $\psi \rightarrow \phi \in [\wedge, \rightarrow, \neg]^n$ corresponds to a subset in $ucv^n(\phi)$.

For every subset $X \subset ucv^n(\phi)$ there is a formula $\psi \in [\wedge, \rightarrow, \neg]^n$ such that $\alpha^n(\psi) = \downarrow(ucv^n(\phi) \setminus X)$, because $Exm([\wedge, \rightarrow, \neg]^n)$ is the exact model of $[\wedge, \rightarrow, \neg]^n$. Infer that $\alpha^n(\psi \rightarrow \phi) = \downarrow X$ and hence every subset of $ucv^n(\phi)$ corresponds to an equivalence class representable by a formula of the form $\psi \rightarrow \phi$. $\dashv$

The following theorem is a simple generalization of the technique used in [Bruijn 75a] to calculate the number of equivalence classes in $[\rightarrow]^3$.

3.5.1.31. Theorem. *Let $N(n, 0) = 0$ and $N(n, k) = 2^{|\bigcap\{ucv^n(p_i)|i \leq k\}|}$ for $k > 0$. Moreover, let $M(n, 0) = 2^{|ucv^n(\bot)|}$ and $M(n, k) = 2^{|ucv^n(\bot) \cap \bigcap\{ucv^n(p_i|i \leq k\}|}$. Then:*

$$|Diag([\rightarrow, \neg]^n)| = M(n, 0) + \sum_{k=1}^{n}(-1)^{k-1}\binom{n}{k}(N(n, k) - M(n, k)).$$

Proof. According to lemma 3.5.1.26 and lemma 3.5.1.30 every formula in $[\rightarrow, \neg]^n$ corresponds exactly to a subset of $ucv^n(\bot)$ or $ucv^n(p)$ for some $p \in \{p_1, \ldots, p_n\}$. In general these upper carrier valuations are not disjoint. So, in order to count their subsets, we have to use the rule $|\mathcal{P}(A) \cup \mathcal{P}(B)| = |\mathcal{P}(A)| + |\mathcal{P}(B)| - |\mathcal{P}(A \cap B)|$. So:

$$\begin{aligned} |Diag([\rightarrow, \neg]^n)| \quad = \quad & |\mathcal{P}(ucv^n(\bot))| + \\ & |\bigcup\{\mathcal{P}(ucv^n(p_i)) \mid i \leq n\}| - \\ & |\mathcal{P}(ucv^n(\bot)) \cap \bigcup\{\mathcal{P}(ucv^n(p_i)) \mid i \leq n\}| \end{aligned}$$

Using the symmetry in $Exm([\wedge, \rightarrow, \neg]^n)$, we have

$$|\bigcup\{\mathcal{P}(ucv^n(p_i)) \mid i \leq n\}| = \sum_{k=1}^{n} (-1)^{k-1} \binom{n}{k} N(n, k)$$

and

$$|\mathcal{P}(ucv^n(\bot)) \cap \bigcup\{\mathcal{P}(ucv^n(p_i)) \mid 1 \leq i \leq n\}| = \sum_{k=1}^{n} (-1)^{k} \binom{n}{k} M(n, k).$$

From which the equation follows. ⊣

3.5.1.32. Corollary. *The number of elements in* $[\rightarrow, \neg]^2$ *is:*

$$2^4 + 2(2^8 - 2^2) - (2^2 - 2) = 518.$$

Proof. In $Exm([\wedge, \rightarrow, \neg]^2)$ (see figure 20) we have $ucv^2(\bot) = \{1, 3, 4, 15\}$, $ucv^2(p) = \{4, 5, 6, 7, 8, 9, 10, 15\}$ and $ucv^2(q) = \{2, 3, 10, 11, 12, 13, 14, 15\}$. So we can calculate $ucv^2(\bot) \cap ucv^2(p) = \{4, 15\}$, $ucv^2(p) \cap ucv^2(q) = \{10, 15\}$ and $ucv^2(\bot) \cap ucv^2(p) \cap ucv^2(q) = \{15\}$. According to theorem 3.5.1.31, then $|Diag([\rightarrow, \neg]^2| = 2^4 + 2(2^8 - 2^2) - (2^2 - 2)$ ⊣

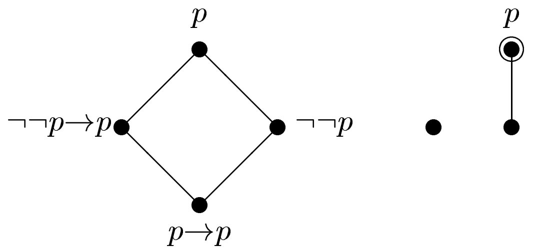
The 518 elements of the diagram of $[\rightarrow, \neg]^2$ have been calculated using the model $Exm([\wedge, \rightarrow, \neg]^n)$. They are listed in appendix B.1.

Applying the method of theorem 3.5.1.31 on $Exm([\wedge, \rightarrow, \neg]^3)$, Renardel de Lavalette calculated the cardinality of $Diag([\rightarrow, \neg]^3)$.

3.5.1.33. Fact. $|Diag([\rightarrow, \neg]^3)| = 3.2^{2\,148} - 546$

3.5.2 The $[\wedge, \rightarrow, \neg\neg]$ fragments

The fragment $[\wedge, \rightarrow, \neg\neg]^n$ does have an exact model which is not an exact Kripke model. This is even true if $n = 1$, see figure 22, where the irreducible classes in the diagram correspond to the formulas $\neg\neg p$ and $\neg\neg p \rightarrow p$. The exact model needs a Kripke completion to force $\neg\neg p$ in the appropriate node.



22. FIGURE. *The diagram of* $[\wedge, \rightarrow, \neg\neg]^1$ *and its universal model. (The encircled node has been added.)*

As we will see, for each n there is universal model for $[\wedge, \rightarrow, \neg\neg]^n$ that is a simple Kripke extension of the exact model of $[\wedge, \rightarrow, \neg\neg]^n$.

As $[\wedge, \rightarrow, \neg\neg]^n$ is a subfragment of $[\wedge, \rightarrow, \neg]^n$, the following fact is a simple consequence of theorem 3.5.0.4.

3.5.2.34. FACT. *The fragment* $[\wedge, \rightarrow, \neg\neg]^n$ *is complete for finite* $\cap$*-independent* n*-models.*

Obviously, a node k in a finite $\cap$-independent n-model with $atom^n(k) = \{p_1, \ldots, p_n\}$ will force every formula in $[\wedge, \rightarrow, \neg\neg]^n$.

3.5.2.35. DEFINITION. *A finite* $\cap$*-independent* n*-model* K *is a* proper $[\wedge, \rightarrow, \neg\neg]^n$ model *if for every* $k \in K$ *with* $\delta(k) > 0$*, there is a* $l > k$ *such that* $atom^n(l) = \{p_1, \ldots, p_n\}$

In this subsection $Th^n(k)$ will denote the set of formulas in $[\wedge, \rightarrow, \neg\neg]^n$ forced by k.

3.5.2.36. LEMMA. *The fragment* $[\wedge, \rightarrow, \neg\neg]^n$ *is complete for proper* $[\wedge, \rightarrow, \neg\neg]^n$ *models.*

Proof. We will prove that for $\phi, \psi \in [\wedge, \rightarrow, \neg\neg]^n$ such that $\phi \nvdash \psi$, there is a k in a proper $[\wedge, \rightarrow, \neg\neg]^n$ model with $k \Vdash \phi$ and $k \nVdash \psi$.

Let $\phi, \psi \in [\wedge, \rightarrow, \neg\neg]^n$ and $\phi \nvdash \psi$. According to fact 3.5.2.34, there is a k in a $\cap$-independent n-model with $k \Vdash \phi$ and $k \nVdash \psi$. By induction on the depth of k we will prove that we can extend the submodel $\uparrow k$ to a proper $[\wedge, \rightarrow, \neg\neg]^n$ model, without changing the $[\wedge, \rightarrow, \neg\neg]^n$ theory of k.

If $\delta(k) = 0$, then $\uparrow k$ is already a proper $[\wedge, \rightarrow, \neg\neg]^n$ model. For the induction step, add a terminal node k_n to $\uparrow k$, such that $atom^n(k_n) = \{p_1, \ldots, p_n\}$ and for all $l \geq k$ with $l \notin Ter(k)$, $l > k_n$. Using induction on the length of formula $\chi \in [\wedge, \rightarrow, \neg\neg]^n$ it is straightforward to prove that $k \Vdash \chi$ iff k forces χ in the extended model. From which we conclude that the $[\wedge, \rightarrow, \neg\neg]^n$ theory of k in both models is the same. $\dashv$

The semantic types in $[\wedge, \rightarrow, \neg\neg]^n$ will be defined as the semantic types of $[\wedge, \rightarrow, \neg]^n$ restricted to proper $[\wedge, \rightarrow, \neg\neg]^n$ models.

3.5.2.37. DEFINITION. *Let* k *be a node in a proper* $[\wedge, \rightarrow, \neg\neg]^n$ *model then* $\tau^n(k)$*, the* semantic type *of* k *in* $[\wedge, \rightarrow, \neg\neg]^n$ *is defined by:*

$$\tau^n(k) = \langle atom^n(k), \{\tau^n(l) \mid k < l\}\rangle.$$

If t and t' are semantic types in $[\wedge,\rightarrow,\neg]^n$ then define $t \preceq t'$ if $t = t'$ or $t' \in j_1(t)$.

The proofs of the following facts are the same as in section 3.5.

3.5.2.38. FACTS. *Let k and l be nodes in proper $[\wedge,\rightarrow,\neg\neg]^n$ models.*

1. $\tau^n(k) = \tau^n(l) \quad \Leftrightarrow \quad k \rightleftarrows^n l$;
2. $\tau^n(k) \preceq \tau^n(l) \quad \Rightarrow \quad Th^n(k) \subseteq Th^n(l)$.

Let us now define the model $Umod([\wedge,\rightarrow,\neg\neg]^n)$, that will be proved in the sequel to be the universal model for $[\wedge,\rightarrow,\neg\neg]^n$.

3.5.2.39. DEFINITION. *We define $Umod([\wedge,\rightarrow,\neg\neg]^n) = \langle T, \preceq, j_0\rangle$, where T is the set of semantic types in $[\wedge,\rightarrow,\neg\neg]^n$.*

3.5.2.40. FACTS.

1. *$Umod([\wedge,\rightarrow,\neg\neg]^n)$ is a proper $[\wedge,\rightarrow,\neg\neg]^n$ model;*
2. *if t a semantic type in $[\wedge,\rightarrow,\neg\neg]^n$ then $\tau^n(t) = t$ in $Umod([\wedge,\rightarrow,\neg\neg]^n)$;*
3. *$Umod([\wedge,\rightarrow,\neg\neg]^n)$ is complete for $[\wedge,\rightarrow,\neg\neg]^n$: if ϕ and ψ in $[\wedge,\rightarrow,\neg\neg]^n$ we have*
$$\phi \vdash \psi \quad \Leftrightarrow \quad [\![\phi]\!] \subseteq [\![\psi]\!].$$

As proper $[\wedge,\rightarrow,\neg\neg]^n$ models are finite $\cap$-independent n-models, we may use $Newatom^n(k)$, $\Delta Newatom^n(k)$ as defined in definition 3.5.0.20. Observe that, as $Diag([\wedge,\rightarrow,\neg\neg]^n)$ is obviously finite, the following definition is allowed.

3.5.2.41. DEFINITION. *For a node k in a proper $[\wedge,\rightarrow,\neg\neg]^n$ model define $\phi^n(k)$, the* type *of k in $[\wedge,\rightarrow,\neg\neg]^n$ as:*

$$\phi^n(k) = \bigwedge Th^n(k).$$

3.5.2.42. LEMMA. *Let k and l be nodes in proper $[\wedge,\rightarrow,\neg\neg]^n$ models. If $l \Vdash \phi^n(k)$ then $\tau^n(k) \preceq \tau^n(l)$ or $atom^n(l) = \{p_1,\ldots,p_n\}$.*

Proof. Assume $l \Vdash \phi^n(k)$ and $atom^n(l) \neq \{p_1,\ldots,p_n\}$. To prove $\tau^n(k) \preceq \tau^n(l)$ we will use induction on $\delta(l)$, the depth of l. If $\delta(l) = 0$ then $k \nVdash \bigwedge atom^n(l) \wedge \Delta Newatom^n(l) \rightarrow \bigwedge\{p_1,\ldots,p_n\}$ and hence there is a $k' > k$ such that $k' \Vdash \bigwedge atom^n(l) \wedge \Delta Newatom^n(l)$ and $k' \nVdash \bigwedge\{p_1,\ldots,p_n\}$. In a proper $[\wedge,\rightarrow,\neg\neg]^n$ model we may infer that k' has to be a terminal node and hence $\tau^n(k') = \tau^n(l)$. Which proves $\tau^n(k) \preceq \tau^n(l)$.

If $\delta(l) > 0$, let $l' > l$. Then $l' \Vdash \phi^n(k)$ and $\delta(l') < \delta(l)$. According to the induction hypothesis we will have $\tau^n(k) \preceq \tau^n(l')$. This proves that $j_1(\tau^n(l)) \subseteq j_1(\tau^n(k))$. As obviously the assumption implies that $atom^n(k) \subseteq atom^n(l)$, we may infer that $\tau^n(k) \preceq \tau^n(l)$. $\dashv$

3.5.2.43. COROLLARY. *Let k and l be nodes in proper $[\wedge,\rightarrow,\neg\neg]^n$ models, then:*

$$\tau^n(k) \preceq \tau^n(l) \quad \Leftrightarrow \quad Th^n(k) \subseteq Th^n(l) \quad \Leftrightarrow \quad l \Vdash \phi^n(k).$$

To prove *Umod*($[\wedge, \rightarrow, \neg\neg]^n$) to be an universal model for $[\wedge, \rightarrow, \neg\neg]^n$ we will use the next lemma. We will write k_n to refer to the node in *Umod*($[\wedge, \rightarrow, \neg\neg]^n$) with $atom^n(k_n) = \{p_1, \ldots, p_n\}$.

3.5.2.44. LEMMA. *Let X be a closed subset in Umod($[\wedge, \rightarrow, \neg\neg]^n$), containing k_n. Define $\phi^n(X) = \bigwedge\bigcap\{Th^n(l) \mid l \in X\}$. Then for every node k in Umod($[\wedge, \rightarrow, \neg\neg]^n$):*

$$k \in X \quad \Leftrightarrow \quad k \Vdash \phi^n(X).$$

Proof. That $k \in X$ implies $k \Vdash \phi^n(X)$ is clear from the definition of $\phi^n(X)$. For the other direction, like in theorem 3.5.0.17 we proceed by induction over $\delta(k)$, the depth of k.

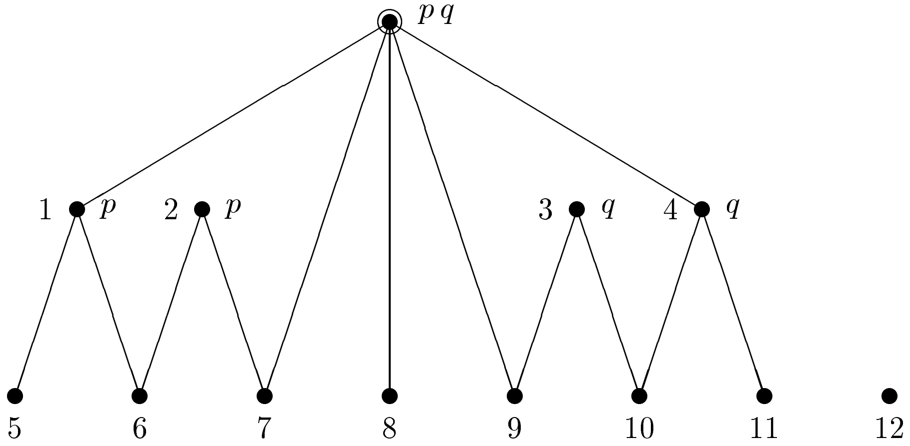
So assume $k \Vdash \phi^n(X)$. If $\delta(k) = 0$, then $\Delta Newatom^n(k) \vdash p \rightarrow q$ for every p and q in $\{p_1, \ldots, p_n\}$. Either $k = k_n$ and $k \in X$ by definition, or $\phi^n(X) \nvdash \bigwedge atom^n(k) \wedge \Delta Newatom^n(k) \rightarrow \bigwedge\{p_1, \ldots, p_n\}$. Suppose $k \neq k_n$. Then for some $l \in X$ there is a $l' \geq l$ such that $l' \Vdash \phi^n(X) \nvdash \bigwedge atom^n(k) \wedge \Delta Newatom^n(k)$ and $l' \nVdash \bigwedge\{p_1, \ldots, p_n\}$. As *Umod*($[\wedge, \rightarrow, \neg\neg]^n$) is a proper $[\wedge, \rightarrow, \neg\neg]^n$ model, this implies that l' is a terminal node and $atom^n(k) = atom^n(l)$. As semantic types are unique in *Umod*($[\wedge, \rightarrow, \neg\neg]^n$), infer that $k = l'$. The set X was supposed to be closed, so, from $l \leq k$ we conclude that $k \in X$.

If $\delta(k) > 0$ then for $k' > k$ we conclude from $k' \Vdash \phi^n(X)$ and the induction hypothesis that $k' \in X$. As *Umod*($[\wedge, \rightarrow, \neg\neg]^n$) is a proper $[\wedge, \rightarrow, \neg\neg]^n$ model, there is a $q \in atom^n(k) \setminus \bigcap\{atom^n(l) \mid k < l\}$. Note that for $k < l$ we have $l \Vdash \phi^n(k) \rightarrow q$ but $k \nVdash \phi^n(k) \rightarrow q$. Now suppose that $l \in X$ and $l \Vdash \phi^n(k)$ then by lemma 3.5.2.42 we have $k \leq l$. Hence $(\phi^n(k) \rightarrow q) \in X$ iff $k \notin X$. As $k \Vdash \phi^n(X)$ infer that $k \in X$. ⊣

3.5.2.45. THEOREM. *Umod($[\wedge, \rightarrow, \neg\neg]^n$) is a universal model for $[\wedge, \rightarrow, \neg\neg]^n$).*

Proof. *Umod*($[\wedge, \rightarrow, \neg\neg]^n$) is a complete model for $[\wedge, \rightarrow, \neg\neg]^n$, according to fact 3. By lemma 3.5.2.44 we have an exact correspondence between equivalence classes in $[\wedge, \rightarrow, \neg\neg]^n$ and closed subsets of *Umod*($[\wedge, \rightarrow, \neg\neg]^n$) that include the node k_n. Deleting k_n from *Umod*($[\wedge, \rightarrow, \neg\neg]^n$) and assigning the empty set to $\bigwedge\{p_1, \ldots, p_n\}$, we have an exact correspondence between members of *Diag*() and closed subsets in the resulting model. Clearly *Umod*($[\wedge, \rightarrow, \neg\neg]^n$) is a minimal complete model for $[\wedge, \rightarrow, \neg\neg]^n$. ⊣

3.5.2.46. COROLLARY. *The exact model of $[\wedge, \rightarrow, \neg\neg]^n$ is (isomorphic to) Umod($[\wedge, \rightarrow, \neg\neg]^n$), after deleting k_n.*



23. FIGURE. *The model $Umod([\wedge, \to, \neg\neg]^n)$. The encircled node has been added to the exact model of* $[\wedge, \to, \neg\neg]^2$.

The exact model has 676 upward closed subsets, corresponding to the 676 equivalence classes of $[\wedge, \to, \neg\neg]^2$.

The type formulas in $[\wedge, \to, \neg\neg]^2$ are:

1. $p \wedge \neg\neg q$	5. $\neg\neg q \wedge ((p \to q) \to p)$	9. $(q \to \neg\neg p) \to (p \wedge q)$
2. $p \wedge (\neg\neg q \to q)$	6. $(\neg\neg q \to (p \wedge q)) \to p)$	10. $(\neg\neg p \to (p \wedge q)) \to q)$
3. $q \wedge (\neg\neg p \to p)$	7. $(p \to \neg\neg q) \to (p \wedge q)$	11. $\neg\neg p \wedge ((q \to p) \to q)$
4. $\neg\neg p \wedge q$	8. $(p \to q) \wedge (q \to p) \wedge \neg\neg p$	12. $((q \to p) \to \neg\neg p) \to (p \wedge q)$

3.5.3 The $[\to, \neg\neg]$ fragments

The diagram of $[\to, \neg\neg]^n$ is not a lattice (it does not have a bottom element) if $n > 1$. For $n = 1$ we have, of course, $Diag([\to, \neg\neg]^1) \cong Diag([\wedge, \to, \neg\neg]^1)$ (see figure 22).

To calculate the diagram of $[\to, \neg]^n$ we will use the universal model of $[\wedge, \to, \neg\neg]^n$.

3.5.3.47. LEMMA. *Every formula in $[\wedge, \to, \neg\neg]^n$ is equivalent to a conjunction of formulas in $[\to, \neg\neg]^n$*

Proof. The proof is much like that of lemma 3.5.1.25. We proceed by induction on the length of ϕ. Only the cases in which ϕ is a double negation or an implication are non-trivial. If $\phi = \neg\neg\psi$, then according to the induction hypothesis ψ is a conjunction of formulas in $[\to, \neg\neg]^n$. Now apply the **IpL** theorem $\neg\neg(A \wedge B) \equiv \neg(A \to \neg B)$ to show that ϕ is equivalent to a formula in $[\to, \neg]^n$.

In the case that $\phi = \psi \to \chi$, the proof runs like in 3.5.1.25. $\dashv$

3.5.3.48. LEMMA. *An **IpL** formula ϕ is equivalent to a formula in $[\to, \neg\neg]^n$ iff $\phi \equiv \psi \to p$ for some $\psi \in [\wedge, \to, \neg\neg]^n$ and $p \in \{p_1, \ldots, p_n\}$.*

Proof. The proof is essentially the same as that of lemma 3.5.1.26. Observe that double negations can be treated as implications, using $\neg\neg\phi \equiv (\neg\neg\phi \to \phi) \to \phi$. $\dashv$

As in subsection 3.5.1, for the calculation of the number of classes in $Diag([\rightarrow,\neg\neg]^n)$ it is more convenient to work with the complement of $[\![\phi]\!]$ in $Umod([\wedge,\rightarrow,\neg\neg]^n)$.

3.5.3.49. DEFINITION. *Let $[\![\phi]\!]$ be the valuation of formulas in $Umod([\wedge,\rightarrow,\neg\neg]^n)$. Define $\alpha^n(\phi) = Umod([\wedge,\rightarrow,\neg\neg]^n) \setminus [\![\phi]\!]$.*

The proof of the following facts is exactly the same as for lemma 3.5.1.28.

3.5.3.50. FACTS. *Let ϕ and ψ be formulas in $[\wedge,\rightarrow,\neg\neg]^n$. Then*

1. $\alpha^n(\phi) \subseteq \alpha^n(\psi) \quad \Leftrightarrow \quad \psi \vdash \phi$;
2. $\alpha^n(\phi \wedge \psi) = \alpha^n(\phi) \cup \alpha^n(\psi)$;
3. $\alpha^n(\phi\rightarrow\psi) = \downarrow(\alpha^n(\psi) \setminus \alpha^n(\phi))$;
4. $\alpha^n(\neg\neg\phi) = \downarrow Umod([\wedge,\rightarrow,\neg\neg]^n) \setminus (\downarrow Umod([\wedge,\rightarrow,\neg\neg]^n) \setminus \ dar\alpha^n(\phi)) = \downarrow Umod([\wedge,\rightarrow,\neg\neg]^n) \setminus \downarrow[\![\phi]\!]$.

3.5.3.51. DEFINITION. *For a formula ϕ in $[\wedge,\rightarrow,\neg\neg]^n$ we define $ucv^n(\phi)$,* the upper carrier *of ϕ, as the set of maximal elements in $\alpha^n(\phi)$.*

Observe that the element k_n in $Umod([\wedge,\rightarrow,\neg\neg]^n)$, with $atom^n(k_n) = \{p_1,\ldots,p_n\}$, is in $[\![\phi]\!]$ for every $\phi \in [\wedge,\rightarrow,\neg\neg]^n$ and hence in no $\alpha^n(\phi)$ or $ucv^n(\phi)$.

3.5.3.52. LEMMA. *For $\phi \in [\wedge,\rightarrow,\neg\neg]^n$ let $An^n(\phi)$ be the set of equivalence classes in $[\wedge,\rightarrow,\neg\neg]^n$ that have a representative of the form $\psi\rightarrow\phi$ with $\psi \in [\wedge,\rightarrow,\neg\neg]^n$. Then*

$$|An^n(\phi)| = |\mathcal{P}(ucv^n(\phi))| = 2^{|ucv^n(\phi)|}.$$

Proof. The proof is essentially the same as for lemma 3.5.1.30. ⊣

3.5.3.53. THEOREM.

$$|Diag([\rightarrow,\neg\neg]^n)| = \sum_{k=1}^{n}(-1)^{k-1}\binom{n}{k}N(n,k).$$

where $N(n,k) = 2^{|\bigcap\{ucv^n(p_i)|i\leq k\}|}$.

Proof. The proof is a simplified version of the proof of theorem 3.5.1.31, using the symmetry in $Umod([\wedge,\rightarrow,\neg\neg]^n)$. ⊣

3.5.3.54. COROLLARY. *The number of elements in $[\rightarrow,\neg\neg]^2$ is:*

$$2.2^7 - 2^2 = 252.$$

Proof. In $Umod([\wedge,\rightarrow,\neg\neg]^n$ (see figure 23) we have $ucv^2(p) = \{3,4,5,6,7,8,12\}$ $ucv^2(q) = \{1,2,8,9,10,11,12\}$. So we have $ucv^2(p) \cap ucv^2(q) = \{8,12\}$. According to theorem 3.5.3.53, then $|Diag([\rightarrow,\neg]^2| = 2.2^7 - 2^2$ ⊣

Applying the method of theorem 3.5.1.31 on $Exm([\wedge,\rightarrow,\neg]^3)$, Renardel de Lavalette calculated the cardinality of $Diag([\rightarrow,\neg\neg]^3)$.

3.5.3.55. FACT. $|Diag([\rightarrow,\neg\neg]^3)| = 3.2^{689} - 380$

3.6 The $[\wedge,\to]$ fragments

The $[\wedge,\to]^n$ fragments of **IpL** are much like the $[\wedge,\to,\neg]^n$ fragments[4].

As we will see, the only difference between the semantic types in $[\wedge,\to,\neg]^n$ and $[\wedge,\to]^n$ is that in the later the semantic types with $j_0(k)=\{p_1,\ldots,p_n\}$ are redundant. Observe that a node with such a semantic type (i.e. where all the atoms hold), forces all formulas in $[\wedge,\to]^n$.

3.6.0.1. DEFINITION. *A finite $\cap$-independent n-model K is a* proper $[\wedge,\to]^n$ model *if for no $k\in K$ we have $atom^n(k)=\{p_1,\ldots,p_n\}$.*

Any $\cap$-independent n-model can easily be turned into a proper $[\wedge,\to]^n$ model.

3.6.0.2. DEFINITION. *Let K be a $\cap$-independent n-model. Then K^- is the model resulting from K after leaving out all nodes k with $atom^n(k)=\{p_1,\ldots,p_n\}$.*

3.6.0.3. LEMMA. *Let K be a finite $\cap$-independent n-model and $k\in K^-$. Then for all $\phi\in[\wedge,\to]^n$:*

$$k\Vdash_K\phi\quad\Leftrightarrow\quad k\Vdash_{K^-}\phi.$$

Proof. Let us use $\Vdash'$ for forcing in K^- in contrast to $\Vdash$ for forcing in K. We proceed by induction on the length of ϕ. The cases where ϕ is either atomic or a conjunction are trivial. So let $\phi=\psi\to\chi$. Suppose $k\Vdash\phi$ and let $l\in K^-$ such that $k\le l$ and $l\Vdash'\psi$. Using the induction hypothesis we conclude that $l\Vdash\psi$ and hence $l\Vdash\chi$. Again by the induction hypothesis, we infer that $l\Vdash'\chi$. Which proves $\forall l\ge k(l\Vdash'\psi\quad\Rightarrow\quad l\Vdash'\chi)$, i.e. $k\Vdash'\phi$.

Now suppose $k\Vdash'\phi$ and $l\in K$ with both $k\le l$ and $l\Vdash\psi$. If $atom^n(l)=\{p_1,\ldots,p_n\}$ then l forces all formulas of $[\wedge,\to]^n$ and hence also $l\Vdash\chi$. Otherwise, we have $l\in K^-$. By the induction hypothesis $l\Vdash'\psi$ and, as $k\Vdash'\phi$, also $l\Vdash'\chi$. Again with the induction hypothesis, we conclude $l\Vdash\chi$. Which proves $k\Vdash\phi$. $\dashv$

The following theorem justifies our definition of proper $[\wedge,\to]^n$ models.

3.6.0.4. THEOREM. *The fragment $[\wedge,\to]^n$ is complete for proper $[\wedge,\to]^n$ models.*

Proof. Let ϕ and ψ be formulas in $[\wedge,\to]^n$, such that $\phi\nvdash\psi$. As $[\wedge,\to]^n$ is a subfragment of $[\wedge,\to,\neg]^n$, by application of theorem 3.5.0.4, there is a node k in a $\cap$-independent n-model K with $k\Vdash\phi$ and $k\nVdash\psi$. From $k\nVdash\psi$ infer that $k\in K^-$. As K^- is a proper $[\wedge,\to]^n$ model and according to lemma 3.6.0.2 $k\Vdash\phi$ and $k\Vdash\psi$ in K^-. $\dashv$

3.6.0.5. DEFINITION. *For a node k in a proper $[\wedge,\to]^n$ model define the* semantic type *of k in $[\wedge,\to]^n$ as:*

$$\tau^n(k)=\langle atom^n(k),\{\tau^n(l)\mid k<l\}\rangle.$$

[4] As an alternative notation of $[\wedge,\to,\neg]^n$ we could have taken $[\wedge,\to,\perp]^n$.

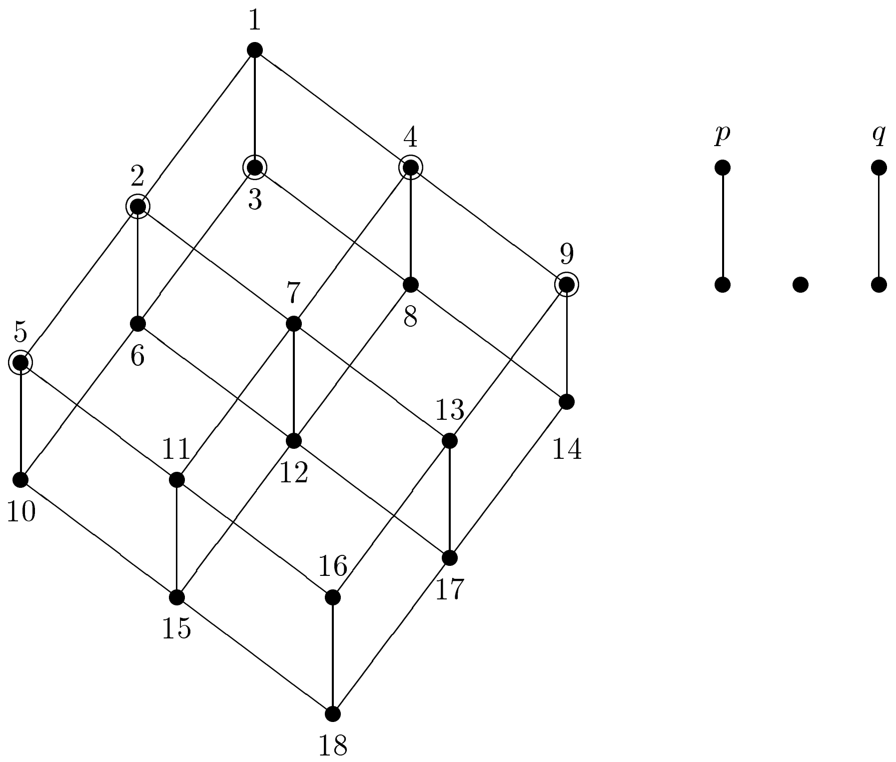
Semantic types in $[\wedge, \rightarrow]^n$ are a special case of semantic types in $[\wedge, \rightarrow, \neg]^n$ and they are ordered in the same way. Obviously there are only finitely many semantic types in $[\wedge, \rightarrow]^n$.

For the proof of the following facts one only has to modify slightly the corresponding proofs in section 3.5. Obviously $Th^n(k)$ in this section means the theory of node k in $[\wedge, \rightarrow]^n$.

3.6.0.6. FACTS. *Let k and l be nodes in proper $[\wedge, \rightarrow]^n$ models.*

1. $\tau^n(k) = \tau^n(l) \quad \Leftrightarrow \quad k \rightleftarrows^n l$;
2. $\tau^n(k) \preceq \tau^n(l) \quad \Rightarrow \quad Th^n(k) \subseteq Th^n(l)$.

3.6.0.7. DEFINITION. *If T is the set of semantic types in $[\wedge, \rightarrow]^n$, then define $Exm([\wedge, \rightarrow]^n) = \langle T, \preceq, j_0 \rangle$.*



24. FIGURE. *The fragment $[\wedge, \rightarrow]^2$ and the model $Exm([\wedge, \rightarrow]^n)$.*

The formulas in $Diag([\wedge, \rightarrow]^2)$:

1. $p \wedge q$	10. $q \to p$
2. p	11. $(p \to q) \to q$
3. $(p \to q) \wedge (q \to p)$	12. $(((p \to q) \to p) \to p) \wedge (((q \to p) \to q) \to q)$
4. q	13. $(q \to p) \to p$
5. $(p \to q) \to p$	14. $p \to q$
6. $((p \to q) \to q) \to p$	15. $((q \to p) \to q) \to q$
7. $((p \to q) \to q) \wedge ((q \to p) \to p)$	16. $((p \to q) \wedge (q \to p)) \to p$
8. $((q \to p) \to p) \to q$	17. $((p \to q) \to p) \to p$
9. $(q \to p) \to q$	18. $p \to p$

As in section 3.5 the following facts are rather simple consequences of the definition of $Exm([\wedge, \to]^n)$.

3.6.0.8. FACTS.

1. *$Exm([\wedge, \to]^n)$ is a proper $[\wedge, \to]^n$ model;*
2. *for t a semantic type in $[\wedge, \to]^n$ we have $\tau^n(t) = t$ in $Exm([\wedge, \to]^n)$;*
3. *$Exm([\wedge, \to]^n)$ is complete for $[\wedge, \to]^n$: for ϕ and ψ in $[\wedge, \to]^n$ we have*

$$\phi \vdash \psi \quad \Leftrightarrow \quad [\![\phi]\!] \subseteq [\![\psi]\!].$$

As $Diag([\wedge, \to]^n)$ is finite, the following definition is allowed.

3.6.0.9. DEFINITION. *For a node k in a proper $[\wedge, \to]^n$ model define $\phi^n(k)$, the* type *of k in $[\wedge, \to]^n$ as*

$$\phi^n(k) = \bigwedge Th^n(k).$$

3.6.0.10. LEMMA. *Let k and l be nodes in proper $[\wedge, \to]^n$ models. If $l \Vdash \phi^n(k)$ then $\tau^n(k) \preceq \tau^n(l)$.*

Proof. Assume $l \Vdash \phi^n(k)$. We will use induction on $\delta(l)$, the depth of l. If $\delta(l) = 0$ then l is a terminal node. We may conclude that the formula $\bigwedge atom^n(l) \to \bigwedge\{p \to \bigwedge\{p_1, \ldots, p_n\} \mid p \in \{p_1, \ldots, p_n\} \setminus atom^n(l)\}$ does not belong to $Th^n(k)$. Hence for some terminal node k' with $k \leq k'$ we have $atom^n(k') = atom^n(l)$, which proves $\tau^n(l) = \tau^n(k)$.

If $\delta(l) > 0$, let $l' > l$. Then $l' \Vdash \phi^n(k)$ and $\delta(l') < \delta(l)$. According to the induction hypothesis we will have $\tau^n(k) \preceq \tau^n(l')$. This proves that $j_1(\tau^n(l)) \subseteq j_1(\tau^n(k))$. As the assumption implies that $atom^n(k) \subseteq atom^n(l)$, this proves $\tau^n(k) \preceq \tau^n(l)$. $\dashv$

3.6.0.11. COROLLARY. *If k and l are nodes in proper $[\wedge, \to]^n$ models then:*

$$\tau^n(k) \preceq \tau^n(l) \quad \Leftrightarrow \quad Th^n(k) \subseteq Th^n(l) \quad \Leftrightarrow \quad l \Vdash \phi^n(k).$$

3.6.0.12. THEOREM. *The model $Exm([\wedge, \to]^n)$ defined above is the exact Kripke model of $[\wedge, \to]^n$.*

Proof. $Exm([\wedge, \to]^n)$ is complete for $[\wedge, \to]^n$ according to fact 3.6.0.8. We still have to prove that every closed subset X in this model corresponds to a formula in $[\wedge, \to]^n$. The proof is very much like that of theorem 3.5.0.17.

Let $\phi^n(X) = \bigwedge\bigcap\{Th^n(k) \mid k \in X\}$. Then clearly, by definition, it will be true that $X \subseteq [\![\phi^n(X)]\!]$.

To prove the inclusion in the other direction, suppose that $k \Vdash \phi^n(X)$. With induction on $\delta(k)$, the depth of k, we will prove that $k \in X$. If $\delta(k) = 0$ then k is a terminal node and apparently it is the case that for some $l \in X$ we had $l \nVdash \Phi(k)$, where $\Phi(k) = \bigwedge atom^n(k) \rightarrow \bigwedge\{p \rightarrow \bigwedge\{p_1, \ldots, p_n\} \mid p \in \{p_1, \ldots, p_n\} \setminus atom^n(k)\}$.

As in the proof of lemma 3.6.0.10 infer that for some $l \in X$ there is a terminal node $l' > l$ with $atom^n(l) = atom^n(k)$. As the semantic types in $Exm([\wedge, \rightarrow]^n)$ are unique, we conclude that $k = l'$. From $l \in X$ and $l < k$ infer that $k \in X$ as X is a closed subset of $Exm([\wedge, \rightarrow]^n)$.

If $\delta(k) > 0$ then for $k' > k$ we conclude from $k' \Vdash \phi^n(X)$ and the induction hypothesis that $k' \in X$. As $Exm([\wedge, \rightarrow]^n)$ is a proper $[\wedge, \rightarrow]^n$ model, there is a $q \in atom^n(k) \setminus \bigcap\{atom^n(l) \mid k < l\}$. Note that for $k < l$ we have $l \Vdash \phi^n(k) \rightarrow q$ but $k \nVdash \phi^n(k) \rightarrow q$. Now suppose that $l \in X$ and $l \Vdash \phi^n(k)$ then by lemma 3.6.0.10 we have $k \leq l$. Hence $(\phi^n(k) \rightarrow q) \in X$ iff $k \notin X$. As $k \Vdash \phi^n(X)$ infer that $k \in X$. ⊣

Like $Exm([\wedge, \rightarrow, \neg]^n)$, the model $Exm([\wedge, \rightarrow]^n)$ can stagewise be constructed as the minimal proper $[\wedge, \rightarrow]^n$ model realizing all semantic types in $[\wedge, \rightarrow]^n$. Let us define the $n+1$ stages E_i^n needed in the construction. Recall that $\mathcal{P}^*(X)$ is the set of closed subsets in X.

3.6.0.13. DEFINITION. *Define E_0^n as the set of 2^n terminal nodes with semantic type $\langle Q, \emptyset \rangle$ such that $Q \subset \{p_1, \ldots, p_n\} \neq Q$.*

Now inductively define:

$$E_{m+1}^n = E_m^n \cup \{\langle Q, S \rangle \mid S \in \mathcal{P}^*(E_m^n) \text{ and } Q \subset \bigcap\{j_0(t) \mid t \in S\} \neq Q\}.$$

Where the order in E_m^n is the order of types in $[\wedge, \rightarrow]^n$.

Note that the construction of E_m^n is only possible for $m \leq n$. From the construction of E_n^n the following facts are obvious:

3.6.0.14. FACTS.

1. *E_n^n is a proper $[\wedge, \rightarrow]^n$ model;*
2. *Every semantic type of $[\wedge, \rightarrow]^n$ is realized in E_n^n exactly once;*
3. *$E_n^n = Exm([\wedge, \rightarrow]^n)$.*

Let us return to the type formulas in $[\wedge, \rightarrow]^n$.

3.6.0.15. DEFINITION. *Let k be a node in a proper $[\wedge, \rightarrow]^n$ model and $X \subset \{p_1, \ldots, p_n\}$. Define:*

1. *$Newatom^n(k) = \{q \mid q \in \bigcap\{atom^n(l) \mid k < l\} \setminus atom^n(k)\}$;*
2. *$\Delta X = \bigwedge\{p \rightarrow q \mid p, q \in X\}$;*
3. *$\psi^n(k) = \phi^n(k) \rightarrow q$, where $q \in Newatom^n(k)$;*
4. $\psi^n(k) = \begin{cases} \phi^n(k) \rightarrow \bigwedge\{p_1, \ldots, p_n\} & \text{if } \delta(k) = 0 \\ \\ \phi^n(k) \rightarrow q, \text{ where } q \in Newatom^n(k) & \text{otherwise.} \end{cases}$

The proper definition of $\psi^n(k)$ of course requires a choice of $q \in Newatom^n(k)$. As this choice will not make any difference in the sequel one may take for example the p_i with the least i such that $p_i \in Newatom^n(k)$. If k is a terminal node, by defining $\bigcap\emptyset = \{p_1,\ldots,p_n\}$, we have $Newatom^n(k) = \{p_1,\ldots,p_n\} \setminus atom^n(k)$.

3.6.0.16. LEMMA. *If k and l nodes in proper $[\wedge,\to]^n$ models then:*

$$l \nVdash \psi^n(k) \quad\Leftrightarrow\quad \tau^n(l) \preceq \tau^n(k).$$

Proof. If k is a terminal node, the lemma is rather trivial. So, assume $\delta(k) > 0$. To prove $l \nVdash \psi^n(k) \;\Rightarrow\; \tau^n(l) \preceq \tau^n(k)$, let $l \nVdash \psi^n(k)$. As $\psi^n(k) = \phi^n(k)\to q$, this implies, for some $l' \geq l$, that $l' \Vdash \phi^n(k)$ and $l' \nVdash q$, where $q \in Newatom^n(k)$. According to corollary 3.5.0.16, $l' \Vdash \phi^n(k)$ implies $\tau^n(k) \preceq \tau^n(l')$. In finite $\cap$-independent models, it is not difficult to prove that if $\tau^n(k) \prec \tau^n(m)$ (i.e. $\tau^n(k) \preceq \tau^n(m)$ but $\tau^n(k) \neq \tau^n(m)$), then $m \Vdash q$, for $q \in Newatom^n(k)$. As $l' \nVdash q$ and obviously from $l \leq t'$ we may conclude that $\tau^n(l) \preceq \tau^n(l')$, we have $\tau^n(l) \preceq \tau^n(l') = \tau^n(k)$.

To prove $\tau^n(l) \preceq \tau^n(k) \;\Rightarrow\; l \nVdash \psi^n(k)$, observe that by definition $k \nVdash q$. So, if $\tau^n(l) \preceq \tau^n(k)$, then $l \Vdash \psi^n(k)$ would imply, by corollary 3.5.0.16, that $k \Vdash \psi^n(k)$. As $k \Vdash \phi^n(k)$, we would have $k \Vdash q$, a contradiction. Hence, we conclude $l \nVdash \psi^n(k)$. $\dashv$

We are now ready for a characterization of $\phi^n(k)$, the type of k in $[\wedge,\to,\neg]^n$. An analogous characterization was used, as a definition, in [De Jongh 68] (also in [De Jongh 70], [De Jongh 80] and [JHR 91]). We will use the exact model of $[\to,\neg]^n$ in the characterization.

3.6.0.17. DEFINITION. *If k is a node in $Exm([\wedge,\to]^n)$ and $q \in Newatom^n(k)$ then define:*

$$\Phi^n(k) = \begin{cases} \bigwedge atom^n(k) \wedge \Delta Newatom^n(k) & \text{if } \delta(k)=0 \\[1em] \bigwedge atom^n(k) \wedge \Delta Newatom^n(k) \wedge \\ \quad \bigwedge\{\psi^n(l)\to q \mid k <_1 l\} \wedge \\ \quad \bigwedge\{\psi^n(m) \mid not\,(m \leq k) \text{ and} \\ \qquad \bigcap\{atom^n(l) \mid k < l\} \subseteq atom^n(m)\} & \text{if } \delta(k) > 0. \end{cases}$$

3.6.0.18. THEOREM. *If k is a node in $Exm([\wedge,\to]^n)$ then $\phi^n(k) \equiv \Phi^n(k)$.*

Proof. If k is a terminal node, first observe that trivially $k \Vdash \Phi^n(k)$ as $Exm([\wedge,\to]^n)$ is a proper $[\wedge,\to]^n$ model. If $l \in Exm([\wedge,\to]^n)$ and $l \Vdash \Phi^n(k)$ then clearly $atom^n(k) \subseteq atom^n(l)$ and, again because $Exm([\wedge,\to]^n)$ is a proper $[\wedge,\to]^n$ model, for no $l' \geq l$ it will be true that $l' \Vdash p$ for some $p \notin atom^n(k)$. Hence l has to be a terminal node with $atom^n(l) = atom^n(k)$, which proves $\tau^n(k) = \tau^n(l)$, so $k = l$.

So assume $\delta(k) > 0$. To prove $\phi^n(k) \vdash \Phi^n(k)$ we show that $k \Vdash \Phi^n(k)$. That $k \Vdash \bigwedge atom^n(k) \wedge \Delta Newatom^n(k) \wedge$ is rather obvious. For $k < l$ we have $l \Vdash q$ and $k \nVdash \psi^n(l)$ by lemma 3.6.0.16. According to the same lemma $k \Vdash \psi^n(m)$ if not $m \leq k$, which proves that k will also force the last of the conjunctions in $\Phi^n(k)$.

For the proof of the other direction, assume $l \Vdash \Phi^n(k)$. We will show that as a consequence $k \leq l$ and hence $l \Vdash \phi^n(k)$. As $Exm([\wedge,\to]^n)$ is the exact model of $[\wedge,\to]^m$, this proves $\Phi^n(k) \vdash \phi^n(k)$.

Suppose $Newatom^n(k) \subseteq atom^n(l)$. Then, using the last part in the conjunction of $\Phi^n(k)$, not $k \leq l$ implies $\Phi^n(k) \vdash \psi^n(l)$. As $l \nVdash \psi^n(l)$, infer that $k \leq l$ and hence $l \Vdash \phi^n(k)$.

If $Newatom^n(k)$ is not a subset of $atom^n(l)$, then $l \nVdash q$ for every $q \in Newatom^n(k)$ (because $l \Vdash \Delta Newatom^n(k)$). Hence if $k <_1 k'$ then, using the third conjunction in $\Phi^n(k)$, we have $l \nVdash \psi^n(k')$. By lemma 3.6.0.16 this implies that $l \leq k'$. Hence $atom^n(l)$ will be included in $atom^n(k) \cup Newatom^n(k)$. From $atom^n(k) \subseteq atom^n(k')$ and $Newatom^n(k) \cap atom^n(l) = \emptyset$ infer that $atom^n(l) = atom^n(k)$ and hence $\tau^n(k) = \tau^n(l)$. As semantic types are unique in $Exm([\wedge,\to]^n)$, we conclude $k = l$ and $l \Vdash \phi^n(k)$. ⊣

We may now use the exact model of the fragment $[\wedge,\to]^n$ to prove a characterization of the $[\wedge,\to]$ formulas in **IpL**. Recall the definition of K^- from definition 3.6.0.2.

3.6.0.19. Theorem. *If ϕ is an* **IpL** *formula, then ϕ is equivalent to a $[\wedge,\to]$ formula if for every node k in a finite Kripke model:*

$$k \Vdash \phi \quad \Leftrightarrow \quad ((\uparrow k)^{\cap})^{-} \Vdash \phi.$$

Proof. For $\phi \in [\wedge,\to]$ we have by theorem 3.5.0.3 that $k \Vdash \phi \quad \Leftrightarrow \quad (\uparrow k)^{\cap}) \Vdash \phi$. Using theorem 3.6.0.3 we may infer that $k \Vdash \phi \quad \Leftrightarrow \quad ((\uparrow k)^{\cap})^{-} \Vdash \phi$.

To prove the other direction, let ϕ be a formula in $\mathbf{IpL}^n$ with the property that for every finite Kripke model K and every node $k \in K$, $k \Vdash \phi \quad \Leftrightarrow \quad ((\uparrow k)^{\cap})^{-} \Vdash \phi$. Let $\chi \in [\wedge,\to,\neg]^n$ be the formula with $[\![\chi]\!] = [\![\phi]\!]$ in $Exm([\wedge,\to]^n)$. For a node k in a proper $[\wedge,\to]$ model we have, using fact 3.6.0.6.1: $k \Vdash \phi \quad \Leftrightarrow \quad k \Vdash \chi$. Hence we have:

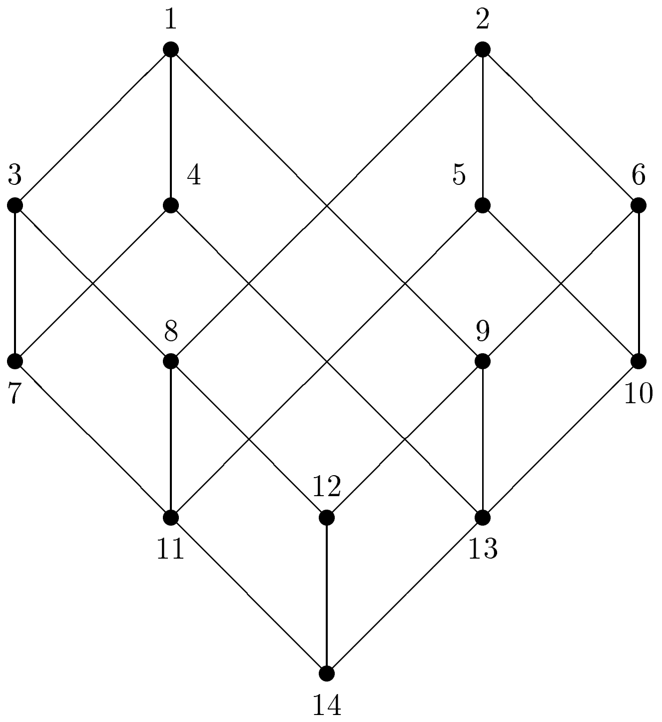
$$k \Vdash \phi \quad \Leftrightarrow \quad (\uparrow k)^{\cap} \Vdash \phi \quad \Leftrightarrow \quad (\uparrow k)^{\cap} \Vdash \chi \quad \Leftrightarrow \quad k \Vdash \chi.$$

Which proves $\phi \equiv \chi$. ⊣

3.6.1 The $[\to]$ fragments

The $[\to]$ fragments are the most expressive fragments in **IpL** with only one connective. For example $[\to]^3$ has 25 165 802 equivalence classes, whereas the fragments with three atoms and exactly one of the other connectives in $\{\wedge,\vee,\neg,\neg\neg\}$ all have less then 10 classes.

To calculate the diagram of $[\to]^n$ we have to use the exact Kripke model of $[\wedge,\to]^n$, for example, as $Diag([\to]^n)$ for $n > 1$ is not a lattice and hence does not have an exact model of its own.



25. FIGURE. *The diagram of* $[\rightarrow]^2$.

The formulas in $Diag([\rightarrow]^2)$:

1. p	6. $(q\rightarrow p)\rightarrow q$	11. $((q\rightarrow p)\rightarrow q)\rightarrow q$
2. q	7. $q\rightarrow p$	12. $(((p\rightarrow q)\rightarrow q)\rightarrow p)\rightarrow p$
3. $(p\rightarrow q)\rightarrow p$	8. $(p\rightarrow q)\rightarrow q$	13. $((p\rightarrow q)\rightarrow p)\rightarrow p$
4. $((p\rightarrow q)\rightarrow q)\rightarrow p$	9. $(q\rightarrow p)\rightarrow p$	14. $p\rightarrow p$
5. $((q\rightarrow p)\rightarrow p)\rightarrow q$	10. $p\rightarrow q$	

To calculate the number of classes in $Diag([\rightarrow]^n)$ we will proceed much like in subsection 3.5.1. The proofs of the following lemma's, preparing for theorem 3.6.1.26, are omitted, as they are essentially the same as for the lemma's 3.5.1.25 up to 3.5.1.30.

3.6.1.20. LEMMA. *Every formula in* $[\wedge,\rightarrow]^n$ *is equivalent to a conjunction of formulas in* $[\rightarrow]^n$

3.6.1.21. LEMMA. *An* **IpL** *formula* ϕ *is equivalent to a formula in* $[\rightarrow]^n$ *iff* $\phi\equiv\psi\rightarrow p$ *for some* $\psi\in[\wedge,\rightarrow]^n$ *and* $p\in\{p_1,\ldots,p_n\}$.

As in subsection 3.5.1, in the calculation of the number of equivalence classes in $[\wedge,\rightarrow]^n$ it is more convenient to work with the dual of $[\![\phi]\!]$ in $Exm([\wedge,\rightarrow]^n)$.

3.6.1.22. DEFINITION. *Let* $[\![\phi]\!]$ *be the valuation of formulas in* $Exm([\wedge,\rightarrow]^n)$. *Define* $\alpha^n(\phi)=Exm([\wedge,\rightarrow]^n)\setminus[\![\phi]\!]$.

3.6.1.23. LEMMA. *Let ϕ and ψ be formulas in $[\wedge,\rightarrow]^n$. Then*

1. $\alpha^n(\phi) \subseteq \alpha^n(\psi) \quad \Leftrightarrow \quad \psi \vdash \phi$;
2. $\alpha^n(\phi \wedge \psi) = \alpha^n(\phi) \cup \alpha^n(\psi)$;
3. $\alpha^n(\phi{\rightarrow}\psi) = {\downarrow}(\alpha^n(\psi) \setminus \alpha^n(\phi))$.

3.6.1.24. DEFINITION. *For a formula ϕ in $[\wedge,\rightarrow]^n$ we define $ucv^n(\phi)$,* the upper carrier of ϕ, *as the set of maximal elements in $\alpha^n(\phi)$.*

3.6.1.25. LEMMA. *For $\phi \in [\wedge,\rightarrow]^n$ let $An^n(\phi)$ be the set of equivalence classes in $[\wedge,\rightarrow]^n$ that have a representative of the form $\psi{\rightarrow}\phi$, with $\psi \in [\wedge,\rightarrow]^n$. Then*

$$|An^n(\phi)| = |\mathcal{P}(ucv^n(\phi))| = 2^{|ucv^n(\phi)|}.$$

3.6.1.26. THEOREM. *The number of equivalence classes in $[\rightarrow]^n$ is:*

$$\sum_{k=1}^{n}(-1)^{k-1}\binom{n}{k}N(n,k)$$

where $N(n,k) = 2^{|\bigcap\{ucv^n(p_i)|i\leq k\}|}$.

Proof. As in the case of theorem 3.5.1.31, we have to calculate the number of different subsets in the $ucv^n(p_i)$, a union of non-disjunct subsets. The summation above uses the symmetry in $Exm([\wedge,\rightarrow]^n)$. ⊣

3.6.1.27. COROLLARY. *The number of elements in $[\rightarrow]^3$ is:*

$$3.2^{23} - 3.2^3 + 1.2 = 25\,165\,802.$$

Proof. Use $Exm([\wedge,\rightarrow]^3)$ and determine $ucv^3(p)$, $ucv^3(q)$ and $ucv^3(r)$ and their intersections, to calculate $N(3,1) = 23$, $N(3,2) = 3$ and $N(3,3) = 1$. The corollary is a result of the substitution of these values in the formula of theorem 3.6.1.26. ⊣

Applying theorem 3.6.1.26 on $Exm([\wedge,\rightarrow]^4)$, Renardel de Lavalette calculated the cardinality of $Diag([\rightarrow]^4)$.

3.6.1.28. FACT. $|Diag([\rightarrow]^4)| = 2^{623\,662\,965\,552\,393} - 50\,331\,618$

Chapter 4

Restricted nesting of implication in IpL

4.1 Introduction

In Chapter 2 we introduced fragments of modal logic with restricted nesting of $\Box$ and showed how in the hierarchy of fragments $\mathbf{K}^n_m$ the types and semantic types of nodes in finite Kripke models could be defined. Semantic types were used to construct exact Kripke models of the fragments $\mathbf{K}^n_m$.

In **IpL** we will introduce a similar stratification of fragments $\mathbf{IpL}^n_m$ to obtain exact Kripke models. With the exception of $\mathbf{IpL}^1$ (and $\mathbf{IpL}^0$, which is the trivial fragment of the classes $\top$ and $\bot$), an exact model for $\mathbf{IpL}^n$ cannot exist (fact 2.5.0.15 in Chapter 2).

In the sequel we will show that restricting the nesting of implication to a maximum of m and confining the propositional variables to the $\{p_1, \ldots, p_n\}$ yields fragments $\mathbf{IpL}^n_m$ with a finite exact Kripke model.

4.2 Preliminaries

4.2.0.1. Definition. *The level of* nesting of the implication, $\mu(\phi)$, *of an* **IpL** *formula ϕ is defined inductively as:*

- $\mu(p) = 0$ *if p is an atomic formula;*
- $\mu(\psi \circ \chi) = \max\{\mu(\psi), \mu(\chi)\}$ *if* $\circ \in \{\wedge, \vee\}$*;*
- $\mu(\neg\psi) = \mu(\psi) + 1$*;*
- $\mu(\psi \rightarrow \chi) = \max\{\mu(\psi), \mu(\chi)\} + 1$.

The fragment $\mathbf{IpL}^n_m$ is defined as the fragment of **IpL** formulas ϕ with propositional variables restricted to $\{p_1, \ldots, p_n\}$, such that $\mu(\phi) \leq m$.

4.3 Semantic types in $\mathbf{IpL}^n_m$

The definition of a semantic type in $\mathbf{IpL}^n_m$ much resembles the definition in modal logic in Chapter 2.

4.3.0.1. DEFINITION. *Let K be a finite* **IpL** *Kripke model. For $k \in K$ define inductively:*

1. $\tau^n_0(k) = \langle atom^n(k), \emptyset\rangle$;;
2. $\tau^n_{m+1}(k) = \langle atom^n(k), \{\tau^n_m(l) \mid k \leq l\}\rangle$.

As in previous chapters, we define $Th^n_m(k) = \{\phi \in \mathbf{IpL}^n_m \mid k \Vdash \phi\}$.

4.3.0.2. THEOREM. *Let K and L be finite* **IpL** *Kripke models. If $k \in K$ and $l \in L$ then:*

$$\tau^n_m(k) = \tau^n_m(l) \quad \Leftrightarrow \quad Th^n_m(k) = Th^n_m(l).$$

Proof. By induction on m. $\Rightarrow$: For $m = 0$ the proof is obvious as we have $atom^n(k) = atom^n(l)$ iff for all $\phi \in \mathbf{IpL}^n_0$: $k \Vdash \phi \quad \Leftrightarrow \quad l \Vdash \phi$. (Note that the fragment $\mathbf{IpL}^n_0$ is the fragment $[\wedge, \vee]^n$ in the previous chapter).

Assume the theorem to be true for m and let $\tau^n_{m+1}(k) = \tau^n_{m+1}(l)$. To prove $Th^n_{m+1}(k) = Th^n_{m+1}(l)$ we will show for all $\phi \in \mathbf{IpL}^n_{m+1}$ we have $k \Vdash \phi \quad \Leftrightarrow \quad l \Vdash \psi$.

Apply induction on the length of ϕ. In case ϕ is atomic, a conjunction or a disjunction the proof that $k \Vdash \phi \quad \Leftrightarrow \quad l \Vdash \phi$ is straightforward. So let $\phi = \psi \rightarrow \chi$ and assume $k \Vdash \psi \rightarrow \chi$. Note that both ψ and χ will be formulas in $\mathbf{IpL}^n_m$.

Let $l \leq h$ and $h \Vdash \psi$. As by definition $\tau^n_m(h) \in j_1(\tau^n_{m+1}(l))$ and $j_1(\tau^n_{m+1}(k)) = j_1(\tau^n_{m+1}(l))$, for some h' such that $k \leq h'$ we have $\tau^n_m(h) = \tau^n_m(h')$. From the first induction hypothesis ($\tau^n_m(k) = \tau^n_m(l) \Rightarrow Th^n_m(k) = Th^n_m(k)$) infer that $h' \Vdash \psi$. As $k \Vdash \psi \rightarrow \chi$ and $k \leq h'$ also $h' \Vdash \chi$. Again by the first induction hypothesis we may conclude that $h \Vdash \chi$. From which we conclude $l \Vdash \psi \rightarrow \chi$.

By interchanging the role of k and l this proof can also be used to prove the other direction: $l \Vdash \phi \quad \Rightarrow \quad k \Vdash \phi$.

As the case that ϕ is a negation is treated likewise, this completes the proof that for all $\phi \in \mathbf{IpL}^n_{m+1}$ we have $k \Vdash \phi \quad \Leftrightarrow \quad l \Vdash \psi$.

$\Leftarrow$: Assume for all $\phi \in \mathbf{IpL}^n_{m+1}$ that $k \Vdash \phi \quad \Leftrightarrow \quad l \Vdash \phi$. We will again apply induction on m to prove $\tau^n_{m+1}(k) = \tau^n_{m+1}(l)$. Obviously $atom^n(k) = atom^n(l)$ and hence the case that $m = 0$ is simple.

For the induction step, assume $k \leq h$. So we may infer that $\tau^n_m(h) \in j_1(\tau^n_{m+1}(k))$. We will prove that for some h' such that $l \leq h'$ it is true that $\tau^n_m(h) = \tau^n_m(h')$. In this way we show that $j_1(\tau^n_{m+1}(k)) \subseteq j_1(\tau^n_{m+1}(l))$. As the proof for the inclusion in the other direction is in fact the same (interchanging k and l) and as $atom^n(k) = atom^n(l)$, this proves $\tau^n_{m+1}(k) = \tau^n_{m+1}(l)$.

As L is a finite model, let $\{l_0, \ldots, l_r\}$ be the finite set of successors of l (including l itself). For each l_i such that $\tau^n_m(l_i) \neq \tau^n_m(h)$, there is, by the induction hypothesis, some formula $\phi_i \in \mathbf{IpL}^n_m$ such that $h \Vdash \phi_i$ or $l \Vdash \phi_i$ but not both.

Define $\Phi = \bigwedge\{\phi_i \mid h \Vdash \phi_i\}$ and $\Psi = \bigvee\{\phi_i \mid h \nVdash \phi_i\}$. Clearly $\Phi \rightarrow \Psi \in \mathbf{IpL}^n_{m+1}$. If $\tau^n_m(h)$ would be different from all $\tau^n_m(l_i)$ then we would have $l \Vdash \Phi \rightarrow \Psi$. So by our assumption that l and k force the same $\mathbf{IpL}^n_m$ formulas, also $k \Vdash \Phi \rightarrow \Psi$. But this would imply $h \Vdash \Phi \rightarrow \Psi$. As obviously $h \Vdash \Phi$ and $h \nVdash \Psi$ this is a contradiction. Hence we may conclude that for some l_i will have the same n, m-type as h. $\dashv$

4.3.0.3. DEFINITION. *Define the order* $\preceq$ *between* n, m*-types as:*

1. $\tau^n_0(k) \preceq \tau^n_0(l)$ *if* $atom^n(k) \subseteq atom^n(l)$*;*
2. $\tau^n_{m+1}(k) \preceq \tau^n_{m+1}(l)$ *if* $\tau^n_{m+1}(k) = \tau^n_{m+1}(l)$ *or* $\tau^n_m(l) \in j_1(\tau^n_{m+1}(k))$.

4.3.0.4. COROLLARY. *Let* K *and* L *be finite* **IpL** *models such that* $k \in K$ *and* $l \in L$*. Then* $\tau^n_m(k) \preceq \tau^n_m(l)$ *implies* $Th^n_m(k) \subseteq Th^n_m(l)$*.*

Proof. Let k' be a new node, having k and l (and hence their successors) as its successors. Moreover let $atom^n(k') = atom^n(k)$. Note that as $\tau^n_m(k) \preceq \tau^n_m(l)$ also $atom^n(k) \subseteq atom^n(l)$ and hence we may take $\uparrow k'$ as a new Kripke model. Obviously $\tau^n_{m+1}(k') = \tau^n_{m+1}(k)$ and $Th^n_{m+1}(k') \subseteq Th^n_{m+1}(l)$. Theorem 4.3.0.2 assures us that $Th^n_{m+1}(k') = Th^n_{m+1}(k)$. $\dashv$

Before defining the type formulas $\phi^n_m(k)$ in $\mathbf{IpL}^n_m$ let us draw some conclusions from this theorem for the structure of the exact Kripke model of $\mathbf{IpL}^n_m$ and give an example.

Let T^n_m be the set of n, m-types in $\mathbf{IpL}^n_m$ and let $Th^n_m(k) = \{\phi \in \mathbf{IpL}^n_m \mid k \Vdash \phi\}$. Obviously T^n_m is finite.

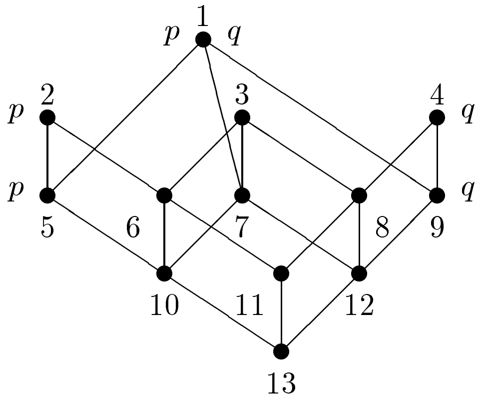
4.3.0.5. DEFINITION. *Define* $Exm(\mathbf{IpL}^n_m)$ *as the Kripke model with* T^n_m *as its domain,* $\preceq$ *as its accessibility relation and* $atom^n(t) = j_0(t)$ *as its valuation.*

4.3.0.6. THEOREM. $Exm(\mathbf{IpL}^n_m)$ *is the exact Kripke model of* $\mathbf{IpL}^n_m$*.*

Proof. Obviously $Exm(\mathbf{IpL}^n_m)$ is a finite **IpL** n-model. By induction on m is easily proved that if $t \in T^n_m$ is an n, m-type, in $Exm(\mathbf{IpL}^n_m)$ we have $\tau^n_m(t) = t$. Hence $Exm(\mathbf{IpL}^n_m)$ is a model realizing exactly all n, m-types in $\mathbf{IpL}^n_m$. $\dashv$

4.3.0.7. COROLLARY. *The exact Kripke model of* $\mathbf{IpL}^n_m$ *is unique up to isomorphism.*

Proof. If M is some exact Kripke model of $\mathbf{IpL}^n_m$ the mapping $\tau^n_m : M \mapsto Exm(bfIpL^n_m)$ is an isomorphism. $\dashv$



26. FIGURE. *$Exm(\mathbf{IpL}^2_1)$, the exact Kripke model of* $\mathbf{IpL}^2_1$.

The irreducible formulas in the exact model of $\mathbf{IpL}^2_1$ are:

1. $p \wedge q$	5. p	9. q	13. $p \to p$
2. $p \wedge \neg q$	6. $\neg q$	10. $q \to p$	
3. $\neg p \wedge \neg q$	7. $p \leftrightarrow q$	11. $\neg(p \wedge q)$	
4. $\neg p \wedge q$	8. $\neg p$	12. $p \to q$	

The exact model of $\mathbf{IpL}^2_1$ was first constructed by Zwanenburg, using the subset of $\wedge$-irreducible formulas in the set of $\vee$-irreducible formulas of $\mathbf{IpL}^2_1$ as a 'skeleton' [Zwanenburg 94].

The exact model can be used to calculate the 98 equivalence classes in the diagram of $\mathbf{IpL}^2_1$, as listed in appendix B.3.

The exact model of $\mathbf{IpL}^2_2$ has 718 elements.

4.4 The n, m-types in IpL

As in case of modal logic we will introduce formulas $\phi^n_m(k)$ for the n, m-type of a node k in a finite Kripke model. We first will define the $\phi^n_m(k)$ and then prove that such a formula is indeed an axiom of $Th^n_m(k)$, the theory of n, m-formulas forced by the node k.

4.4.0.1. DEFINITION. *For a node k in a finite* **IpL** *model, define $\phi^n_m(k)$ inductively as:*

1. $\phi^n_0(k) = \bigwedge atom^n(k)$;
2. $\phi^n_{m+1}(k) =$
   $\bigwedge\{\phi^n_m(l) \to \bigvee\{\phi^n_m(h) \mid k \leq h \text{ and } \phi^n_m(l) \nvdash \phi^n_m(h)\} \mid \tau^n_{m+1}(k) \not\preceq \tau^n_{m+1}(l)\}$.

4.4.0.2. LEMMA. *Let K and L be finite* **IpL** *Kripke models. Assume that $k \in K$ and $l \in L$. Then $\tau^n_m(k) \preceq \tau^n_m(l)$ implies $Th^n_m(k) \subseteq Th^n_m(l)$*

Proof. The case $m = 0$ is obvious. For $m > 0$, by the definition of $\preceq$ we have $\tau^n_m(k) \preceq \tau^n_m$. So apply corollary 4.3.0.4. $\dashv$

4.4.0.3. THEOREM. *Let K and L be finite* **IpL** *Kripke models such that $k \in K$ and $l \in L$. Then:*

$$l \Vdash \phi^n_m(k) \quad \Leftrightarrow \quad \tau^n_m(k) \preceq \tau^n_m(l).$$

Proof. By induction on m. The case $m = 0$ is simple.

So assume $l \Vdash \phi^n_{m+1}(k)$ and let $\tau^n_{m+1}(k) \not\preceq \tau^n_{m+1}(l)$. From the induction hypothesis we know that $l \Vdash \phi^n_m(l)$. By definition of $\phi^n_{m+1}(k)$, infer that $l \Vdash \bigvee\{\phi^n_m(h) \mid k \leq h \text{ and } \phi^n_m(l) \nvdash \phi^n_m(h)\}$.

Hence for some h such that $k \leq h$ we have $l \nVdash \phi^n_m(h)$ and $l \Vdash \phi^n_m(h)$. This is a contradiction as $l \Vdash \phi^n_m(h)$ implies $\phi^n_m(l) \vdash \phi^n_m(h)$. To prove this, take g a node in some **IpL** Kripke model such that $g \Vdash \phi^n_m(l)$. By induction hypothesis $\tau^n_m(l) \preceq \tau^n_m(g)$. So, by the previous lemma, conclude that $g \Vdash \phi^n_m(h)$. Hence, by the completeness theorem for **IpL**, it follows that $\phi^n_m(l) \vdash \phi^n_m(h)$. So $l \Vdash \phi^n_{m+1}(k)$ implies $\tau^n_{m+1}(k) \preceq \tau^n_{m+1}(l)$.

For the other direction, assume that $\tau^n_{m+1} \preceq \tau^n_{m+1}(l)$. We will use the previous lemma and show $k \Vdash \phi^n_{m+1}(k)$ to prove that $l \Vdash \phi^n_{m+1}(k)$.

Let $k \leq h$. If $h \Vdash \phi^n_m(l)$ then, by induction hypothesis, we know $\tau^n_m(l) \preceq \tau^n_m(h)$. Assume $\phi^n_m(l) \Vdash \phi^n_m(h)$. Then $l \Vdash \phi^n_m(h)$ and so, again by induction hypothesis, $\tau^n_m(h) \preceq \tau^n_m(l)$. Hence we would have $Th^n_m(h) = Th^n_m(l)$ and by the theorem in the previous section, $\tau^n_m(h) = \tau^n_m(l)$. Obviously this implies $\tau^n_m(k) \preceq \tau^n_m(l)$.

Hence $h \Vdash \phi^n_m(h)$ and $\phi^n_m(l) \vdash \phi^n_m(h)$. So, for $\tau^n_{m+1}(k) \not\preceq \tau^n_{m+1}(l)$ we have:

$$k \Vdash \phi^n_m(l) \to \bigvee\{\phi^n_m(k) \mid k \leq h \text{ and } \phi^n_m(l) \nvdash \phi^n_m(k)\}.$$

Which we had to prove. ⊣

4.4.0.4. COROLLARY. *For a node k in a finite* **IpL** *Kripke model K the formula $\phi^n_m(k)$ is an axiom of the theory $Th^n_m(k)$.*

Proof. Let $\psi \in Th^n_m(k)$ and assume for some node l in a finite **IpL** Kripke model L that $l \Vdash \phi^n_m(k)$. By the theorem above we have $\tau^n_m(k) \preceq \tau^n_m(l)$ and so $l \Vdash \psi$. ⊣

4.4.0.5. COROLLARY. *If $\phi^n_m(k) \equiv \phi^n_m(l)$ then $\tau^n_m(k) = \tau^n_m(l)$.*

Proof. Obvious, as $Th^n_m(k) = Th^n_m(l)$. ⊣

Chapter 5

Exactly provable L formulas

5.1 Introduction

In this chapter we will study the exactly provable formulas in fragments of provability logic **L** (**GL** in [Boolos 93], **PRL** in [Smoryński 85]). According to Solovay's theorem [Solovay 76] on provability interpretations the theorems of the provability logic **L** are precisely those modal formulas that are provable in **PA** under arbitrary arithmetical interpretations (interpreting $\Box$ as the formalized provability predicate in **PA**). The logic **L** is also known to be the logic of the diagonizable algebras, recently also called Magari algebras. Here, we are concerned with the finitely generated Magari algebras that are embeddable in the Magari algebra of Peano Arithmetic. Shavrukov [Shavrukov 93] characterized these subalgebras, which are recursively enumerable, as having the so-called strong disjunction property.

In the context of the present work the terminology of propositional theories (i.e. sets of propositional modal formulas closed under modus ponens and necessitation) is more convenient.

Let us introduce a new derivability relation to distinguish between the usual modal theories (closed under modus ponens) and the modal theories that are in addition closed under necessitation.

5.1.0.1. Definition. *Let ϕ and ψ be modal formulas. Define:*

$$\phi \mathrel{\dot{\vdash}} \psi \quad \Leftrightarrow \quad \phi \wedge \Box\phi \vdash \psi.$$

A propositional theory in **L** will here be a set of propositional formulas closed under $\dot{\vdash}$. Rephrased in the terminology of propositional theories, we study those theories T over **L** in a finite number of propositional variables that are (faithfully) interpretable in **PA**. Theories correspond to τ-filters in the free Magari algebras and interpretability to embeddability as a subalgebra. Interpretable theories T in $p_1, \cdots, p_n$ are those propositional theories in $p_1, \cdots, p_n$ for which there is a sequence of arithmetical sentences $A_1, \cdots, A_n$ such that an **L** formula ψ is an $\dot{\vdash}$ consequence of T iff ψ^* is a theorem of **PA** in the arithmetical interpretation $*$ in which the atomic formula p_i is interpreted as A_i (see e.g. [Solovay 76], [Boolos 93] or [Smoryński 85]). Written out: T axiomatizes an arithmetically interpreted theory:

$$\{\psi \mid T \vdash \psi\} = \{\psi \mid \vdash_{PA} \psi^*(A_1, \cdots, A_n)\}.$$

The faithfully interpretable propositional theories T in $\mathbf{L}^n$ (i.e., $\mathbf{L}$ restricted to the language of $p_1, \cdots, p_n$) are according to Shavrukov the consistent recursively enumerable (r.e.) theories that satisfy *the strong disjunction property*: $T \vdash \Box\psi \vee \Box\chi$ implies $T \vdash \psi$ or $T \vdash \chi$. (Parenthetically: interpretable theories in infinitely many propositional variables need not be r.e.) The strong disjunction property may be thought of as being composed out of the simple disjunction property: $T \vdash \Box\psi \vee \Box\chi$ implies $T \vdash \Box\psi$ or $T \vdash \Box\chi$, and *ω-consistency*: $T \vdash \Box\psi$ implies $T \vdash \psi$.

An older concept to which this can be related is the concept of *exact provability* introduced in [De Jongh 82] (see also [JC 95]): in the terminology used here[1] a formula can be defined to be exactly provable if it axiomatizes an interpretable theory. That means that an exactly provable formula of $\mathbf{L}$ is a formula ϕ which axiomatizes an arithmetically interpreted propositional theory:

$$\{\psi \mid \phi \vdash \psi\} = \{\psi \mid \vdash_{PA} \psi^*(A_1, \cdots, A_n)\}.$$

One of the objects of our research is to get an overview of exactly provable formulas of low complexity aided by computerized calculations. For that purpose the semantic characterizations in terms of Kripke-models and (semantic) types developed in the previous chapters will be applied to interpretable theories and exactly provable formulas. It turns out that an important role is played by *maximal exactly provable formulas*, i.e. exactly provable formulas that are not implied by any other exactly provable formula, and, more in general, by *maximal theories with the* strong disjunction *property*. The characterizations of these concepts discussed in this chapter make heavy use of the relationship between exactly provable formulas in provability logic and sets of finite *types* of modal formulas as introduced in Chapter 2.

This chapter is built up as follows. After a preliminary section 5.2, characterizations of interpretable theories and exactly provable formulas are given in section 5.3. Maximal exactly provable formulas are discussed in section 5.4. In the last section 5.5, it is shown how the theory was applied to calculate the 62 exactly provable formulas in one propositional variable of modal complexity 1, and the 8 maximal ones among them.

5.2 Preliminaries

The *provability logic* $\mathbf{L}$ is the modal propositional logic with as its axioms the ones of classical propositional logic as well as all formulas of the forms $\Box(\phi \to \psi) \to (\Box\phi \to \Box\psi)$ and $\Box(\Box\phi \to \phi) \to \Box\phi$, and the inference rules modus ponens and necessitation. As

[1] In [HJ 96] exactly provable formulas were called *exact* formulas. In the present context this terminology might suggest a connection with exact models which does not exist.

usual $\Diamond\psi$ is defined as $\neg\Box\neg\psi$, and we will use the abbreviation $\boxdot\,\phi$ for the formula $\phi \wedge \Box\phi$. Note that in $\mathbf{L}$ $\phi \vdash \psi$ is equivalent to $\boxdot\,\phi \vdash \psi$.

We say '*ϕ is interderivable with ψ*' and write $\phi \equiv \psi$ for the conjunction of $\phi \vdash \psi$ and $\psi \vdash \phi$. Note that this implies that always $\phi \equiv \boxdot\,\phi$. We reserve the terminology '*ϕ is equivalent to ψ*' for $\vdash \phi \leftrightarrow \psi$.

Propositional theories in $\mathbf{L}^n$ will here be sets of propositional formulas closed under $\vdash$. Such a propositional theory T is called *consistent* if $T \nvdash \bot$.

By its completeness theorem, $\mathbf{L}$ is the logic of all finite, transitive and irreflexive Kripke-models (a proof can be found in [Boolos 93] and [Smoryński 85]).

Recall from Chapter 2 the definition of $\beta(\phi)$, the *modal degree* of a formula ϕ. The definition of semantic types and type formulas in $\mathbf{L}^n_m$ will be essentially the same as in $\mathbf{K}^n_m$ (see Chapter 2).

5.2.0.1. DEFINITION. *Let k be a node in a finite, transitive and irreflexive Kripke model. Then, $\tau^n_m(k)$, the* n, m-type *of k (in $\mathbf{L}$) is defined by:*

- $\tau^n_0(k) = \langle atom^n(k), \emptyset \rangle$;
- $\tau^n_{m+1}(k) = \langle atom^n(k), \{\, \tau^n_m(l) \mid kRl\} \rangle$.

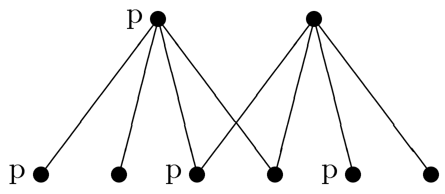
The set of all such n, m-types is written T^n_m. Define $\phi^n_m(k)$, the $\mathbf{L}^n_m$ type *of k inductively as:*

- $\phi^n_0(k) = \phi^n_{\mathbf{CpL}}(k)$
- $\phi^n_{m+1}(k) = \phi^n_{\mathbf{CpL}}(k) \wedge \bigwedge\{\Diamond\phi^n_m(l) \mid kRl\} \wedge \Box\bigvee\{\phi^n_m(l) \mid kRl\}$

One easily verifies that the $n, m+1$-type of k, $\tau^n_{m+1}(k)$ uniquely determines the n, m-type of k. If t is an $n, m+1$-type, let us write $t{\restriction}m$ for the corresponding n, m-type. The following fact will be useful in the sequel of this chapter.

5.2.0.2. FACT. *Let k be a node in a finite, transitive and irreflexive Kripke model. If $m \leq l$ then $\tau^n_m(k) = \tau^n_l(k){\restriction}m$.*

As in that chapter was done for $\mathbf{K}$, the fragment $\mathbf{L}^n_m$ will be the fragment of formulas ϕ in $\mathbf{L}^n$ such that $\beta(\phi) \leq m$. It can be proved that there exists an exact model for each $\mathbf{L}^n_m$.



27. FIGURE. *The construction of ExL^1_0 and ExL^1_1.*

For the infinite fragment $\mathbf{L}^n$ there is no such exact model, but as in case of $\mathbf{K}$ in Chapter 2, there is a canonical (infinite) model ExL^n which is n-complete. It gives considerable insight into the free Magari algebra over n generators.

It is convenient to us to execute most of our constructions inside this model. Many of these constructions are applicable more generally, however.

Let us write $\Box^0\phi = \phi$ and $\Box^{n+1}\phi = \Box\Box^n\phi$. The following facts about the nodes of ExL^n will be useful in the sequel.

5.2.0.3. FACTS.

1. $\delta(k) = m \quad \Leftrightarrow \quad k \Vdash \neg\Box^m\bot \wedge \Box^{m+1}\bot$;
2. *If* $\delta(k), \delta(l) \leq m$ *and* $\tau^n_m(k) = \tau^n_m(l)$, *then* $k = l$;
3. *If* $\delta(k) = m$ *and* $l \Vdash \phi^n_m(k) \wedge \neg\Box^m\bot \wedge \Box^{m+1}\bot$, *then* $k = l$.

These facts suggest a kind of normal form for the irreducible formulas corresponding to the elements of ExL^n.

5.2.0.4. DEFINITION. *Let* $k \in ExL^n$ *and assume* $\delta(k) = m$.
Then $\phi^n(k) \; = \; \phi^n_m(k) \wedge \neg\Box^m\bot \wedge \Box^{m+1}\bot$.

From the n-completeness of ExL^n we conclude that for $k \in ExL^n$ the $\phi^n(k)$ are the irreducible elements in $\mathbf{L}^n$.

5.3 Exactly provable formulas in $\mathbf{L}^n$

As stated in the introduction, Shavrukov's theorem in [Shavrukov 93] gives a characterization of the exactly provable formulas in **L**.

5.3.0.1. FACT. *A formula* $\phi \in \mathbf{L}$ *is exactly provable iff* ϕ *is not a contradiction and has the* strong disjunction property (*is* s.d.):

$$\forall \psi, \chi \in L\,(\phi \vdash \Box\psi \vee \Box\chi \quad \Rightarrow \quad \phi \vdash \psi \quad \text{or} \quad \phi \vdash \chi).$$

The property in this fact is called *steady* by Shavrukov [Shavrukov 93]. Whether a formula $\phi \in \mathbf{L}^n_m$ is steady or not, Shavrukov [Shavrukov 93] also proved, depends only on its behavior with regard to other formulas in $\mathbf{L}^n_m$:

5.3.0.2. FACT. *A formula* $\phi \in \mathbf{L}^n_m$ *is exactly provable iff* ϕ *is not a contradiction and is s.d. for formulas in* $\mathbf{L}^n_m$*:*

$$\forall \psi, \chi \in \mathbf{L}^n_m(\phi \vdash \Box\psi \vee \Box\chi \quad \Rightarrow \quad \phi \vdash \psi \quad \text{or} \quad \phi \vdash \chi)$$

For a simple proof of this last fact see [Zambella 94]. We will transform this characterization of exact provability into a semantic one. This characterization does not work if we are not only interested in exactly provable formulas, but want to study interpretable theories in general (see [HJ 96]).

5.3.0.3. DEFINITION. $\omega^n(\phi) = \{k \in ExL^n \mid k \Vdash \boxdot\phi\}$.
If T *is a propositional theory in* $\mathbf{L}^n$, *then* $\omega^n(T) = \{k \in ExL^n \mid k \Vdash T\}$.

Obviously $\omega^n(\phi)$ and $\omega^n(T)$ will always be *closed upwards* in the sense that, if e.g. $k \in \omega^n(\phi)$ and $k < l$, then $l \in \omega^n(\phi)$.

5.3.0.4. THEOREM. *A formula* $\phi \in \mathbf{L}^n$ *is exactly provable iff* $\omega^n(\phi)$ *is non-empty and* downwards directed, *i.e.* $\forall k, l \in \omega^n(\phi)\, \exists h \in \omega^n(\phi)\, (\, h < k \;\&\; h < l\,)$.

Proof. $\Rightarrow$: Let ϕ be an exactly provable formula in $\mathbf{L}^n$. As ϕ unequals the contradiction by definition, we have $\omega^n(\phi) \neq \emptyset$ by the completeness of ExL^n. To prove the second condition, let $k, l \in \omega^n(\phi)$, and $\phi^n(k), \phi^n(l)$ be (representatives of) the irreducible classes in $\mathbf{L}^n$ corresponding to k and l. Assume that, if $h \in \omega^n(\phi)$ and $h < k$, then $h \not< l$. Then, again by the completeness of ExL^n, we would have $\phi \vdash \Diamond\phi^n(k) \to \Box\neg\phi^n(l)$, or equivalently $\phi \vdash \Box\neg\phi^n(k) \vee \Box\neg\phi^n(l)$. As ϕ is supposed to be exactly provable, ϕ would either prove $\neg\phi^n(k)$ or $\neg\phi^n(l)$, in contradiction with the assumption that $k, l \in \omega^n(\phi)$. Hence, there should be an $h \in \omega^n(\phi)$ such that $h < k$ and $h < l$.
$\Leftarrow$: Let $\psi, \chi \in \mathbf{L}^n$ and $\phi \vdash \Box\psi \vee \Box\chi$, and assume there are $k, l \in \omega^n(\phi)$ such that $k \nVdash \psi$ and $l \nVdash \chi$. By the last condition of the theorem, there is an $h \in \omega^n(\phi)$ such that $h < k$ and $h < l$. As we would then have $h \nVdash \Box\phi \vee \Box\psi$, we obtain a contradiction. Hence, we proved that $\phi \vdash \psi$ or $\phi \vdash \chi$. $\dashv$

By the completeness of $\mathbf{L}$ non-interderivable ϕ and ψ give rise to distinct $\omega^n(\phi)$ and $\omega^n(\psi)$. This is in general not so for theories. An example is the theory axiomatized by p on the one hand, and the theory T_1 axiomatized by $\Box^m\bot \to p$ for each m, on the other. The sets $\omega^1(p)$ and $\omega^1(T_1)$ are the same, consisting of all nodes that together with all their successors force p, but clearly the theories are not: p is not a consequence of T_1. Similarly, the theory $T_2 = \Box^m\bot \to \Box p \vee \Box\neg p$ for each m can be shown to have the strong disjunction property. But not all pairs of nodes in $\omega^1(T_2)$ have a common predecessor in ExL^1, because $\omega^1(T_2)$ consists of those nodes that together with all their successors force p and those nodes that together with all their successors don't force p. This shows that the semantic characterization of exact provability does not generalize to interpretability of non-finitely axiomatizable theories, at least if one doesn't freely use infinite models. For a restricted class of theories that does respect the characterization see [HJ 96].

Note that the $\omega^n(\phi)$ of an exactly provable $\phi \in \mathbf{L}^n$ is infinite by the conditions of the characterization. On the other hand there is a simple correspondence between such an infinite set and a finite set of n, m-types in ExL^n:

5.3.0.5. Definition. *Let ϕ be an $\mathbf{L}^n$ formula.*
Then $T^n_m(\phi) = \{\tau^n_m(k) \mid k \in ExL^n, k \Vdash \boxdot\phi\}$.

5.3.0.6. Lemma. *Let ϕ and ψ be $\mathbf{L}^n_m$ formulas.*
Then $T^n_m(\phi) = T^n_m(\psi)$ iff $\phi \equiv \psi$.

Proof. For the non-trivial direction, from left to right, let $T^n_m(\phi) = T^n_m(\psi)$, and assume $k \Vdash \boxdot\phi$. Then $\tau^n_m(k) \in T^n_m(\phi) = T^n_m(\psi)$. Hence $\tau^n_m(k) = \tau^n_m(k')$ for some k' that forces $\boxdot\psi$. So, $k' \Vdash \psi$ and, since $\psi \in \mathbf{L}^n_m$, $k \Vdash \psi$. The rest is evident. $\dashv$

5.3.0.7. Lemma. *Let ϕ be an $\mathbf{L}^n_{m+k}$ formula.*
Then $T^n_m(\phi) = \{t{\upharpoonright}m \mid t \in T^n_{m+k}(\phi)\}$.

Proof. Obvious, considering fact 5.2.0.2. $\dashv$

5.3.0.8. LEMMA. *For each $\mathbf{L}^n$ formula ϕ and each m there is a finite upwardly closed subset K of $\omega^n(\phi)$ such that the elements of K exactly realize $T^n_m(\phi)$, i.e., $T^n_m(\phi)=\{\tau^n_m(k) \mid k \in K\}$.*

Proof. Just take any finite subset of $\omega^n(\phi)$ such that its elements exactly realize $T^n_m(\phi)$. The upward closure of this set will do, because its elements also force $\boxdot\phi$. ⊣

To find the sets of n, m-types suitable for exactly provable formulas ϕ, we have to translate the conditions on the $\omega^n(\phi)$ of exactly provable ϕ into conditions on the underlying set of n, m-types. For example, for a finite $T^n_m(\phi)$ to correspond to an infinite $\omega^n(\phi)$, it is necessary that some type in $T^n_m(\phi)$ can have itself as a successor. To describe this kind of reflexivity we introduce the notion of a *reflexive type*.

5.3.0.9. DEFINITION. *A type $t \in T^n_{m+1}$ is called* reflexive *if $t{\restriction}m \in j_1(t)$.*

The following theorem is related to lemma 5.13 of [Shavrukov 93].

5.3.0.10. THEOREM. *A formula $\phi \in \mathbf{L}^n_m$ with $m > 0$ is exactly provable iff there is a type $t \in T^n_m(\phi)$ such that $j_1(t) = T^n_{m-1}(\phi)$, which, of course, makes t a reflexive type.*

Proof. $\Rightarrow$: Let $\phi \in \mathbf{L}^n_m$ be an exactly provable formula. Note that $T^n_{m-1}(\phi)$ is a finite set of types. Let $K \subset \omega^n(\phi)$ be finite and closed upwards such that $\{\tau^n_m(k) \mid k \in K\} = T^n_m(\phi)$, as guaranteed to exist by lemma 5.3.0.8. According to theorem 5.3.0.4 we can find an $h \in \omega^n(\phi)$ below all of the elements of K. By lemma 5.3.0.7 this h must have a type as required.
$\Leftarrow$: Assume ϕ and t to fulfill the conditions given. As $T^n_m(\phi) \neq \emptyset$, also $\omega^n(\phi) \neq \emptyset$. Suppose $k, l \in \omega^n(\phi)$. Let K be a finite upwardly closed subset of $\omega^n(\phi)$ such that $k, l \in K$ and $\{\tau^n_{m-1}(k') \mid k' \in K\} = T^n_{m-1}(\phi)$ (compare lemma 5.3.0.8). Consider a world h just below this K such that $\{p \in P^n \mid h \Vdash p\} = j_0(t)$. It will be clear that $\tau^n_m(h) = t$ and (since ϕ is assumed to be an $\mathbf{L}^n_m$ formula) this proves $h \in \omega^n(\phi)$. Of course, $h < k$ and $h < l$, so the conditions of theorem 5.3.0.4 apply to $\omega^n(\phi)$. ⊣

The theory developed in this chapter and in Chapter 2 has enabled us to calculate the exactly provable formulas in $\mathbf{L}^1_1$. This will be explained in more detail in the last section. It will be shown that already in this very first small fragment there are 62 non-interderivable members. It turned out that it was worthwile to single out the 8 maximal elements of these 62.

5.4 Maximal exactly provable formulas

This section will be devoted to maximal exactly provable formulas. First we will have to sharpen our semantic characterization of exactly provable formulas. Let us exploit the relationship between irreducible formulas and semantic types to write $\phi^n_m(t)$ for the $\phi^n_m(k)$ with $\tau^n_m(k) = t$.

5.4.0.1. DEFINITION. *Let C be a set of n, m-types.*
Then $\phi^n_m(C) = \bigvee\{\phi^n_m(t) \mid t \in C\}$.

Recall that $T^n_m(\phi) = \{\tau^n_m(k) \mid k \Vdash \boxdot\phi\}$.

5.4.0.2. Lemma. *If $\phi \in \mathbf{L}^n_m$, then $\phi \equiv \phi^n_m(T^n_m(\phi))$.*

Proof. Immediate from lemma 5.3.0.6 as soon as one realizes that $T^n_m(\phi^n_m(T^n_m(\phi))) = T^n_m(\phi)$. ⊣

5.4.0.3. Lemma. *If $C \subseteq T^n_m$ $(m > 0)$, then $C = T^n_m(\phi)$ for an exactly provable formula $\phi \in \mathbf{L}^n_m$ iff*

1. *There is a finite upwards closed $K \subseteq ExL^n$ such that $C = \{\tau^n_m(k) \mid k \in K\}$ (we will call C* upwards closed realizable*)*
2. *There is $t \in C$ such that $\forall t' \in C\,(t' \restriction (m-1) \in j_1(t))$. Such a type t will be called an* enveloping type *for C.*

Moreover, in that case $\phi \equiv \phi^n_m(C)$.

Proof. $\Rightarrow$: If $\phi \in \mathbf{L}^n_m$ is exactly provable, then $T^n_m(\phi)$ will have the required property 1 by the definition of $T^n_m(\phi)$, property 2 by theorem 5.3.0.10, and satisfies the final requirement by lemma 5.4.0.2.
$\Leftarrow$: We prove that $\phi^n_m(C)$ is an exactly provable formula. To apply theorem 5.3.0.10 to $\phi^n_m(C)$, we have to find an appropriate reflexive n,m-type. By the assumption on C, there is an n,m-type $t \in C$ such that $\forall t' \in C\,(t' \restriction (m-1) \in j_1(t))$. Let K be the upwardly closed realization of the types in C as assumed in the first conditon of this lemma. Note that K realizes precisely the $n,m-1$-types in $\{t' \restriction (m-1) \mid t' \in C\}$ (compare lemma 5.3.0.7). Let k be a (new) root immediately below K such that k forces exactly the elements of $j_0(t)$. Then $\tau^n_m(k) = t$. So, $k \Vdash \boxdot\phi^n_m(C)$ and, hence, t is a member of $T^n_m(\phi^n_m(C)) \subseteq C$ and a type appropriate for the application of theorem 5.3.0.10. ⊣

We will prove that the maximal exactly provable formulas in $\mathbf{L}^n$ correspond to what we will call *tail models* in ExL^n. Clearly this is a result that is, to a large extent, bound to the particular model ExL^n.

5.4.0.4. Definition. *$K \subset ExL^n$ is called a* tail model *iff:*

1. *K is closed upwards;*
2. *there is an $m \in \omega$ such that $\{k \in K \mid \delta(k) \geq m\}$ is linearly ordered by $<$ and all nodes of this set force the same atoms.*

If $k \in ExL^n$, then we write $\uparrow\!k\!\downarrow$ for the tail model consisting of $\uparrow\!k$ and a tail descending from k with the forcing of the atoms as in k.

Our definition of tail model slightly differs from the one in [Visser 84] in that Visser's tail models are equipped with a minimal (infinite-depth) element.

5.4.0.5. Lemma. *If $\phi \in \mathbf{L}^n_m$, $k \in \omega^n(\phi)$ and k has a reflexive n,m-type, then $\uparrow\!k\!\downarrow \subseteq \omega^n(\phi)$.*

Proof. First note that all elements of the tail have the same n,m-type as k. Hence, all these nodes force ϕ, and consequently $\boxdot\,\phi$. $\dashv$

5.4.0.6. LEMMA. *If $K \subset ExL^n$ is a tail model, then $K = \omega^n(\phi)$ for some ϕ in $\mathbf{L}^n$.*

Proof. Let K be $\uparrow\!k\!\downarrow$, k having depth m, and let θ be the conjunction of the propositional variables and negations of propositional variables as they are forced on k. Then $K = \omega^n(\phi)$ for ϕ defined as the conjunction of $\Box^{m+1}\bot \rightarrow \bigvee\{\phi^n(k') \mid k \leq k'\}$ and $\neg\Box^{m+1}\bot \rightarrow \theta \wedge \Diamond\,\phi^n(k)$. $\dashv$

5.4.0.7. LEMMA. *If $\phi \in \mathbf{L}^n$ and $\omega^n(\phi)$ is a tail model, then ϕ is maximal exactly provable.*

Proof. Assume $\omega^n(\phi)$ is a tail model and $\phi \in \mathbf{L}^n_m$. Since $\omega^n(\phi)$ is infinite and $T^n_{m-1}(\phi)$ finite it is obvious that the tail has to contain elements appropriate for an application of theorem 5.3.0.10. This shows that ϕ has to be exactly provable. Assume ψ to be an exactly provable formula such that $\psi \vdash \phi$, i.e., such that $\omega^n(\psi) \subseteq \omega^n(\phi)$. Then, because $\omega^n(\psi)$ is non-empty and downwards directed it has to contain the tail elements from a certain node downwards, and, because it it is closed upwards it has to contain all other elements of $\omega^n(\phi)$, which means that ϕ and ψ are interderivable. Hence, ϕ is maximal exactly provable. $\dashv$

5.4.0.8. LEMMA. *If $\phi \in \mathbf{L}^n$, then there exists a formula $\psi \in \mathbf{L}^n$ such that $\omega^n(\psi) \subseteq \omega^n(\phi)$ and $\omega^n(\psi)$ is a tail model.*

Proof. Let $\phi \in \mathbf{L}^n$ and assume $t \in T^n_m(\phi)$ is a reflexive type with the properties guaranteed to exist by theorem 5.3.0.10. Now, take as in the proof of lemma 5.4.0.3($\Leftarrow$) $k \in \omega^n(\phi)$ with n,m-type t such that $\tau^n_m(\uparrow\!k) = T^n_m(\phi)$. By lemma 5.4.0.5, $\uparrow\!k\!\downarrow \subseteq \omega^n(\phi)$. By lemma 5.4.0.6, there exists a ψ with $\omega^n(\psi) = \uparrow\!k\!\downarrow \subseteq \omega^n(\phi)$. $\dashv$

From lemma 5.4.0.8 it follows immediately that any $\mathbf{L}^n$ formula ϕ is determined uniquely (up to interderivability) by the maximal exactly provable $\mathbf{L}^n$ formulas that imply it. Certainly this does not generalize to interpretable theories. The s.d. theory T_1 axiomatized by $\Box^n\bot \rightarrow p$ for each n that was introduced after theorem 5.3.0.4 provides a counter-example. Its only maximal s.d. extension is the one axiomatized by p. Also, lemma 5.4.0.8 does not, in general imply that ϕ is equivalent to a finite disjunction of maximal exactly provable formulas (each preceded by $\boxdot$), although that may very well be the case. A counter-example is provided by the formula $\top$.

5.4.0.9. THEOREM. *If $\phi \in \mathbf{L}^n$, then ϕ is maximal exactly provable in $\mathbf{L}^n$ iff $\omega^n(\phi)$ is a tail model in ExL^n.*

Proof. The direction from right to left follows from lemma 5.4.0.7. The other direction from 5.4.0.8 using the simple fact that, if one tail model is part of another, they have to be equal. $\dashv$

From lemma 5.4.0.6 and theorem 5.4.0.9 it is clear that there is a one-one correspondence between maximal exactly provable formulas and tail models.

Also from theorem 5.4.0.9, it follows that maximal exactly provable formulas in p cannot be symmetric with regard to p and $\neg p$ as the tail is always asymmetric. We follow with some additional properties and problems concerning maximal exactly provable formulas.

5.4.0.10. THEOREM. *If a formula $\phi \in \mathbf{L}^n_m$ is maximal exactly provable, then there is precisely one reflexive type t in $T^n_m(\phi)$. Moreover, $T^n_{m-1}(\phi) = j_1(t)$.*

Proof. The last part follows immediately from theorem 5.3.0.10. Assume $\phi \in \mathbf{L}^n_m$ with $m > 0$ is maximal exactly provable. Assume s and s' to be two distinct n, m-types in $T^n_m(\phi)$. If k and k' in $\omega^n(\phi)$ realize s and s' respectively, then, by lemma 5.4.0.5, $\uparrow k \downarrow$ and $\uparrow k' \downarrow$ are two distinct tail models within $\omega^n(\phi)$. This contradicts the fact that $\omega^n(\phi)$ is a tail model. $\dashv$

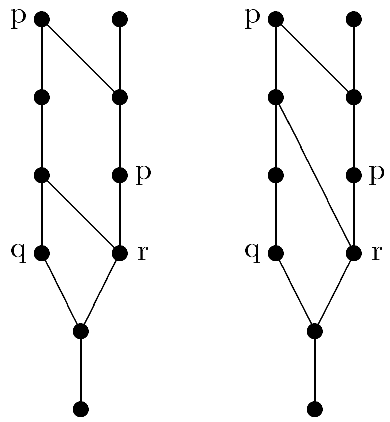
Examples of non-maximal exactly provable $\mathbf{L}^1_1$ formulas with exactly one reflexive $1, 1$-type will be given in the table in the last section.

5.4.0.11. DEFINITION. *An exactly provable $\mathbf{L}^n_m$ formula ϕ is called* n, m-maximal exactly provable *iff, for all exactly provable $\psi \in \mathbf{L}^n_m$ such that $\psi \vdash \phi$, $\psi \equiv \phi$.*

It will turn out in the last section that the $1, 1$-maximal exactly provable formulas in $\mathbf{L}^1$ are maximal exactly provable. In general, however, not all the n, m-maximal exactly provable formulas in $\mathbf{L}^n_m$ are maximal exactly provable. To construct counter-examples the following insight derived from lemma 5.4.0.3 and the fact that, by lemma 5.3.0.6, $\mathbf{L}^n_m$ formulas are, up to $\equiv$, determined by their n, m-types was used.

5.4.0.12. FACT. *The m-maximal exactly provable $\mathbf{L}^n_m$-formulas are the ones with a set of types C that contains exactly one reflexive n, m-type t and for which C is minimal upwardly closed realizable, in the sense that, C is upwardly closed realizable, but this is not the case for any proper subset of C containing t.*

The simplest counter-example we found uses a set of $3, 2$-types with exactly one minimal enveloping type in the sense of the previous fact. Such a set of $3, 2$-types will correspond to a $3, 2$-maximal exactly provable formula. The following two models, both built using only this set of types, show there is a real choice in ordering it.



28. FIGURE. *Two models built from the set of* $3,2$*-types corresponding to a* $3,2$*-maximal exactly provable formula.*

The models above can be extended to tail models corresponding to different maximal exactly provable $\mathbf{L}_3^3$ formulas. From both of these formulas the $3,2$-maximal exactly provable formula corresponding to the set of $3,2$-types is derivable. Hence this $3,2$-maximal exactly provable formula is clearly not maximal exactly provable.

A further conjecture is that the set of n,m-types of an arbitrary exactly provable $\mathbf{L}_m^n$ formula ϕ is the union of the sets of types of the n,m-maximal exactly provable $\mathbf{L}_m^n$ formulas from which ϕ is derivable. That such a union always is the set of types of an exactly provable formula if a common enveloping type is present, follows immediately from the next lemma.

5.4.0.13. LEMMA. *If* C *is the union of sets* $C_1, \cdots, C_k$ *of* n,m *types corresponding to exactly provable* $\mathbf{L}_m^n$ *formulas* $\phi_1, \cdots, \phi_k$ *with an enveloping type* t *for all of* C*, then there exists a* $\phi \in \mathbf{L}_m^n$ *such that* $T_m^n(\phi) = C$*.*

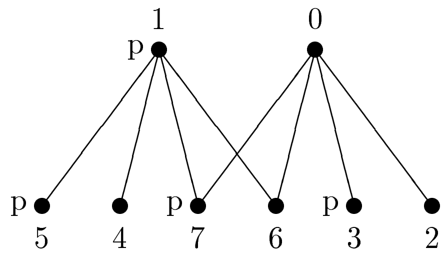
Proof. It suffices to note that, if $K_1, \cdots, K_k$ are upwards closed realizations of $C_1, \cdots, C_k$, then $K_1 \cup \cdots \cup K_k$ is an upwardly closed realization of C, and then to apply lemma 5.4.0.3. ⊣

It is certainly not true that any union of types of n,m-maximal exactly provable formulas is the set of n,m-types of some exactly provable $\mathbf{L}_m^n$ formula. A counter-example is provided by the sets of types belonging to p and to $\neg p$, both $0,1$-maximal exactly provable formulas, which cannot be combined to an exactly provable formula, even for $m=1$. A common enveloping type is needed, and is obviously not available for p and $\neg p$ (see section 5.5).

5.5 Calculating exactly provable formulas

In this section the calculation of the exactly provable formulas in $\mathbf{L}_1^1$ will be discussed. It will be shown that already in this very first small fragment there are 62 non-interderivable members with 8 maximal elements. Of the next fragment $\mathbf{L}_2^1$ even the cardinality of the set of maximal exactly provable elements has eluded us so far. The fragment $\mathbf{L}^1_3$ is definitely too large to attack in this manner.

To calculate the exactly provable formulas in $\mathbf{L}^1_1$ we use sets of $1,1$-types. The $1,1$-types can be ordered into an exact Kripke model $Exm(\mathbf{L}^1_1)$:



29. FIGURE. *An exact Kripke model of* $\mathbf{L}^1_1$.

This exact model corresponds to the first two layers (ExL^1_0 and ExL^1_1) in the construction of ExL^1. In ordering the types into an exact model other choices could have been made, resulting in different models. In fact, in the calculation of exactly provable formulas the choice of the exact model is arbitrary. In the sequel we will denote the $1,1$-types by their number in the exact model above.
In the previous section we proved that $\phi \in L^1_1$ is exactly provable iff

1. $T^1_1(\phi)$ is upwards closed realizable;
2. there is a $t \in T^1_1(\phi)$ such that $\forall t' \in T^1_1(\phi)(t' \upharpoonright 0 \in j_1(t))$.

These criteria are easily translated into a test on a set of $1,1$-types C. The first condition of this test requires C to be upwards closed realizable (in ExL^1 not necessarily in the model above) and the second condition demands an enveloping type in C. Let $\phi \in L^1_1$ and $C = T^1_1(\phi)$. Then ϕ is exactly provable iff

1. if $2 \in C$ or $3 \in C$, then $0 \in C$
   if $4 \in \ \in C$ or $5 \in C$, then $1 \in C$
   if $6 \in C$ or $7 \in C$, then $C \cap \{0,2,4\} \neq \emptyset$ and $C \cap \{1,3,5\} \neq \emptyset$;
2. $6 \in C$ or $7 \in C$ or $C = \{0,2\}$ or $C = \{1,5\}$.

The sets of $1,1$-types corresponding to exactly provable formulas in $\mathbf{L}^1$ can be found in applying the above test to the 255 non-empty subsets of T^1_1. We prefer however to calculate the exactly provable formulas together with their corresponding sets of $1,1$-types. To do so, the exact model above will be used to calculate all $\mathbf{L}^1_1$ formulas in the following manner.

Our computer program generates a list of formulas and sets. It starts with the formulas $\bot$ and p and the sets $[\![\bot]\!] = \emptyset$ and $[\![p]\!] = \{1,3,5,7\}$ (where $[\![\phi]\!] = \{k \in Exm(\mathbf{L}^1_1) \mid k \Vdash \phi\}$).

The list of formulas ϕ and sets $[\![\phi]\!]$ is extended by systematically applying the connectives $(\neg, \wedge, \vee, \rightarrow, \Box)$ and the corresponding set operations, adding a pair consisting of a formula and its set only if the set does not yet occur in the list. In this way we ensure that no two distinct interderivable formulas will occur in the list. Note that this computation of the Lindenbaum algebra of an exact Kripke model is similar to the calculation described in Chapter 2.

In generating the list of formulas and sets the test defined above is applied to distinguish the exactly provable formulas in $\mathbf{L}_1^1$.

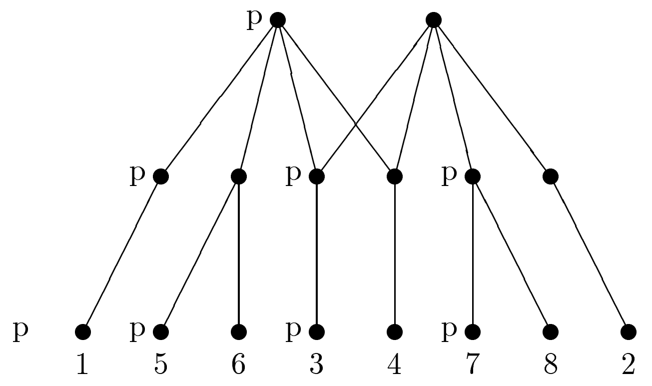
The exactly provable formulas in $\mathbf{L}_1^1$ have been listed in appendix B.4.

To find the 1,1-maximal exactly provable formulas ϕ in the list, one has to look for the minimal sets $T_1^1(\phi)$ (i.e. those that do not occur as a proper subset of some $T_1^1(\psi)$ in the list).

The sets of types of this kind are:

1. $\{1,5\}$	3. $\{0,1,7\}$	5. $\{1,4,7\}$	7. $\{0,3,7\}$
2. $\{0,2\}$	4. $\{0,1,6\}$	6. $\{1,4,6\}$	8. $\{0,3,6\}$

It turns out that each of these 1,1-maximal exactly provable formulas is maximal exactly provable. The corresponding tail models can be found, using the model below, by extending the submodels $\uparrow k$ downward with a tail of copies of k for each of the numbered elements.



30. FIGURE. *Extending* ExL_1^1 *to find tail models.*

We will give these maximal exactly provable formulas in $\mathbf{L}_1^1$ a more informative form:

1. p
2. $\neg p$
3. $(\Box p \to \Box\bot) \wedge (\Box\neg p \to \Box\bot) \wedge (\neg p \to \Box\neg p)$
4. $(\Box p \to \Box\bot) \wedge (\Box\neg p \to \Box\bot) \wedge (p \to \Box p)$
5. $p \leftrightarrow \Diamond\neg p \vee \Box\bot$
6. $p \leftrightarrow \Box\neg p$
7. $p \leftrightarrow \neg\Box p$
8. $p \leftrightarrow \Box\neg p \vee \neg\Box\bot$

Formulas 1 and 2 correspond to *provable* and *refutable* sentences in **PA**. Formulas 6 and 7 can be (faithfully) interpreted by *Gödel-sentences* and their duals in **PA**. Similarly, formulas 3 and 4 correspond to *Rosser-sentences* and their duals in **PA**. The only small surprise is formed by formula 8 and its dual 5. It is easy to see that 8 is interderivable with $p \leftrightarrow \Box\Box\bot \wedge \neg\Box\bot$ and thus, of course, 5 with $p \leftrightarrow (\Box\Box\bot \to \Box\bot)$.

These two formulas are not $\mathbf{L}_1^1$, but can apparently interderivably be given as such. Note also that, by the fixed point theorem of $\mathbf{L}$ (see e.g., [Smoryński 85], [Boolos 93]), there is no surprise in the fact that in the equivalences of 5 and 8 the p in the right hand side can be eliminated in favor of the $\perp$, but only in the fact that by using p instead of $\perp$ one can push down the complexity.

Chapter 6

A family of propositional testers

6.1 Introduction

The common origin of the theorem testers treated here is the semantic tableau method introduced by Beth in 1955 [Beth 55]. Beth defined semantic tableaux both for classical and intuitionistic (predicate) logic. Restricting these methods to propositional formulas yields decision procedures for the classical propositional logic (**CpL**) and the intuitionistic propositional logic (**IpL**). By appropriately changing the rules, the semantic tableau method can also be used in modal logic.

Algorithms to decide for a given logic L and a given formula A whether $\vdash_L A$ are called *formula testers* here, whereas the (usual) term *theorem prover* is reserved for algorithms that produce a proof (for example in natural deduction style) if the given formula is a theorem. In [Hendriks 80] for example, a theorem prover is given, based on the tableau method for **CpL**.

6.2 Preliminaries

A *tableau* is an ordered set of sequents $L \bullet R$, where L and R are structures of sequences of formulas. A *tableau method* defines what shall be considered as a sequent and gives a set of rules to derive new sequents from a given sequent (thus defining the order of the tableau). In those cases where application of a rule results in more than one sequent, the tableau is said to *branch* into subtableaux.

A sequent $L \bullet R$ is *closed* if $L \cap R \neq \emptyset$. Here we used the intersection of L and R as if they were sets. In the sequel we will treat L and R as sets if in the context there is no risk of confusion. Let us write $\#X$ for the number of elements in X and $Sub(X)$ for the set of subformulas of formulas in X.

The rules of a tableau method resemble a system of derivation rules for sequents in a system of sequent calculus. Treatment of a sequent results in a finite directed acyclic graph. If all terminal sequents are closed, the resulting tableau is a proof, but upside down, as the tableau method started with the conclusion of the proof and the closed sequents correspond to axioms.

All formula testers presented here are based on a tableau method. In the rest of this chapter we will use the following convention of writing:

p	an atomic formula
A, B, C	formulas
K, M, N, S, T, U	sequences of formulas, not containing duplicates
L, R	ordered pairs of sequences of formulas

To test whether or not the formula A is derivable, one starts with a sequent $\bullet A$ and applies the rules, until all sequents are either closed or no rule can be applied. If enough sequents close, the tableau is said to *close* and A is derivable. Otherwise the tableau is said to stay *open* and A is not derivable. In the description of the tableaux algorithms we will use rewriting rules on socalled *split sequents* $L \bullet R$, where L and R are finite sequences of finite sequences of formulas, separated by additional symbols (like ; and ,). If $L \bullet R$ is a split sequent, $A, L \bullet R$ is the split sequent where A is added on the left hand side of the leftmost sequence in L. Also we will write $L \bullet R, A$ for adding A to the right of R. However, we will assume that before adding a formula A to a sequence X in a split sequent, a check is performed whether A is already an element of X. So, if $A \in X$ then A, X and X, A are equal to X.

6.3 CpLtest: a CpL tester

The simplest member of our family is *CpLtest*, a formula tester for **CpL**.

The split sequents of *CpLtest* are of the form $M; N \bullet S; T$. To test whether A is derivable, one starts with the split sequent $; \bullet A;$. Hence A is a formula in S, the sequence of righthand side formulas to be treated. The *CpLtest* rules below are applied to a split sequent by treating the leftmost formula of S or the rightmost formula of N. If $N = S = \emptyset$ treatment stops. In treating a formula A subformulas of A are placed in S or in N. A formula in S (N) is placed in sequence T (M) to facilitate recognition of a closure, i.e. a formula A occurring both in L and in R.

The rules of the tester *CpLtest* are:

$$(pR)\ \frac{L \ \bullet\ p, R}{L \ \bullet\ R, p} \qquad\qquad (pL)\ \frac{L, p \bullet ; T}{p, L \bullet ; T}$$

$$(\neg R)\ \frac{L \ \bullet\ \neg A, R}{L, A \ \bullet\ R, \neg A} \qquad\qquad (\neg L)\ \frac{L, \neg A \bullet ; T}{\neg A, L \ \bullet\ A; T}$$

$$(\wedge R)\ \frac{L \ \bullet\ A \wedge B, R}{L \ \bullet\ A, R, A \wedge B \quad L \ \bullet\ B, R, A \wedge B} \qquad\qquad (\wedge L)\ \frac{L, A \wedge B \bullet ; T}{A \wedge B, L, A, B \bullet ; T}$$

$$(\vee R)\ \frac{L\ \bullet\ A\vee B, R}{L\ \bullet\ A, B, R, A\vee B} \qquad (\vee L)\ \frac{L, A\vee B\bullet\,; T}{A\vee B, L, A\bullet\,; T\quad A\vee B, L, B\bullet\,; T}$$

$$(\to R)\ \frac{L\ \bullet\ A\to B, R}{L, A\ \bullet\ B, R, A\to B} \qquad (\to L)\ \frac{L, A\to B\bullet\,; T}{A\to B, L\ \bullet\ A; T\quad A\to B, L, B\bullet\,; T}$$

None of the *CpLtest* rules is applicable to a closed split sequent.

Note that all L-rules require that $S=\emptyset$. So for each split sequent at most one rule is applicable and hence the algorithm *CpLtest* is deterministic. We define measures of complexity that will strictly decrease with each application of a *CpLtest* rule.

6.3.0.1. DEFINITION. *Let X be a set of split sequents.*

1. $\gamma(p)=0$;
2. $\gamma(\neg A)=\gamma(A)+1$;
3. $\gamma(A\circ B)=\gamma(A)+\gamma(B)+2$ if $\circ\in\{\wedge,\vee,\to\}$;
4. $\mu(M;N\ \bullet\ S;T)=\Sigma\{\gamma(A)+1\mid A\in N\}+\Sigma\{\gamma(A)+1\mid A\in S\}$;
5. $\eta(M;N\ \bullet\ S;T)=\#Sub(N)+\#Sub(S)$;
6. $\sigma(X)=\Sigma\{2^{\eta(L\bullet R)}\times\mu(L\bullet R)\mid L\bullet R\in X\}$.

6.3.0.2. LEMMA. *If $L\ \bullet\ R$ is a split sequent then $\mu(L\ \bullet\ R)\geq 0$. If $L\ \bullet\ R$ is a split sequent derived from split sequent $L'\ \bullet\ R'$ by application of one of the* CpLtest *rules, then*

$$\mu(L\ \bullet\ R)<\mu(L'\ \bullet\ R')$$

If X is a set of split sequents then $\sigma(X)\geq 0$. If X' is a set of split sequents derived from X by application of one of the CpLtest *rules (replacing the split sequent treated by the result(s) of the application of the* CpLtest *rule), then*

$$\sigma(X')<\sigma(X)$$

Proof. By checking the rules. ⊣

The measure $\sigma(X)$ provides us with an upper bound to the number of steps it may take *CpLtest* to treat all split sequents in X (and the resulting sequents and so on) until no rule of *CpLtest* is applicable (hence $N=S=\emptyset$).

6.3.0.3. DEFINITION. *A split sequent $L\ \bullet\ R$ is* open *if it is not closed and no* CpLtest *rule is applicable. A split sequent $L\ \bullet\ R$ is* closing *if it is closed or if a* CpLtest *rule is applicable and the resulting split sequent(s) are closing. We will write $L\overset{\bullet}{-}R$ if $L\ \bullet\ R$ is closing.*

Note that the definition of *closing* is sound because the algorithm is terminating.

6.3.0.4. LEMMA. *$L\overset{\bullet}{-}R$ is closing iff $L\vdash\bigvee R$.*

Proof. If $L \bullet R$ is closed then of course $L \vdash \bigvee R$. If $L \bullet R$ is open, define a model M by taking $M \models p \quad \Leftrightarrow \quad p \in L$ for atomic formulas p. Using the fact that all formulas in N and S in the split sequent are treated by one of the rules, one proves $M \models \bigwedge L$ and $M \not\models \bigvee R$.

As the tableau for a split sequent is a finite tree of split sequents, we can proceed by induction on the depth of the sequent (closed split sequents having depth zero).

By checking the *CpLtest* rules, observe that they correspond to equivalent statements about the derivability relation of **CpL** as stated in lemma 6.3.0.4. For example for the $\vee L$-rule one can prove

$$A \vee B \vdash C \quad \Leftrightarrow \quad A \vee B, A \vdash C \quad \text{and} \quad A \vee B, B \vdash C$$

in **CpL**. $\dashv$

We now present *CPLtest* as a pseudo-code program, called *Ctest*. In the pseudo-code language the notation of the sequence operation A, X introduced earlier, will be replaced by $\langle A, X\rangle$, writing A for $\langle A, \emptyset\rangle$ and $\langle A, B, X\rangle$ for $\langle A, \langle B, X\rangle\rangle$. $Ctest(,,\phi,)$, the program *Ctest*, with as its input the formula ϕ, will return the value **true** if $\vdash_{CpL} \phi$ and the value **false** otherwise.

```
Ctest(M, N, S, T : sequence of formula) : bool
  if S ≠ ∅
  then let S = ⟨A, S'⟩
       if A ∈ M ∪ N then true
       else in case A
            atomic : Ctest(M, N, S', ⟨A, T⟩)
            ¬B     : Ctest(M, ⟨B, N⟩, S', ⟨A, T⟩)
            B ∧ C  : if Ctest(M, N, ⟨B, S'⟩, ⟨A, T⟩)
                     then Ctest(M, N, ⟨C, S'⟩, ⟨A, T⟩)
                     else false
            B ∨ C  : Ctest(M, N, ⟨B, C, S'⟩, ⟨A, T⟩)
            B→C    : Ctest(M, ⟨B, N⟩, ⟨C, S'⟩, ⟨A, T⟩)
  else if N ≠ ∅
       then let N = ⟨A, N'⟩
            if A ∈ T then true
            else in case A
                 atomic : Ctest(⟨A, M⟩, N', , T)
                 ¬B     : Ctest(⟨A, M⟩, N', B, T)
                 B ∧ C  : Ctest(⟨A, M⟩, ⟨A, B, N'⟩, , T)
                 B ∨ C  : if Ctest(⟨A, M⟩, ⟨B, N⟩, , T)
                          then Ctest(⟨A, M⟩, ⟨C, N⟩, , T)
                          else false
                 B→C    : if Ctest(⟨A, M⟩, N, B, T)
                          then Ctest(⟨A, M⟩, ⟨C, N⟩, , T)
                          else false
       else false
```

To calculate an upper bound to the amount of time needed to calculate $Ctest(,,\phi,)$, we can make use of the measure σ defined above, as $\sigma(\{;\ \bullet\ \phi;\})$ is an upper bound to the number of calls to the *Ctest* procedure.

6.3.0.5. FACT. *Let $|\phi|$ be the length of formula ϕ, i.e. the number of atoms and connectives in ϕ. Then*

1. $\gamma(\phi) < |\phi|$;
2. $\mu(;\ \bullet\ \phi;) \leq |\phi|$;
3. $\eta(;\ \bullet\ \phi;) < |\phi|$;
4. $\sigma(;\ \bullet\ \phi;) < |\phi|.2^{|\phi|}$.

Next we need an upper bound to the time it takes to respond to a call of *Ctest*. In the worst case the procedure *Ctest* involves the following steps:

1. determine whether $S = \emptyset$ and $N = \emptyset$,
2. splitting a sequence X as $\langle A, X' \rangle$,
3. determine whether a formula is in $M \cup N$ or T,
4. decompose a formula A into its principal subformulas,
5. concatenating a formula A and a sequence X into $\langle X, A \rangle$, which should result in the sequence X if A is already a member of X.

We assume that placing a (new) call to *Ctest* will take a small constant amount of time. Let us assume that X is the largest sequence and D is the longest formula in the *Ctest* call we are dealing with.

The first step will only take a small constant amount of time, as will the second step if sequences are represented as linked lists for example.

To determine equality of two formulas A and B will cost, at the most, $\min\{|A|, |B|\}$ steps. Hence, as an upper bound for the third step we can use $\#X \times |D|$.

Step four can be done in a number of steps linear in the length of the formula treated. Step five may occur thrice and each time we may use $\#X \times |D|$ as an upper bound.

From the rules of *Ctest* it is clear that the original ϕ from the input is the longest formula appearing in any of the consecutive calls to *Ctest*. Hence in the formulas above we can replace D by ϕ. Also, from the rules of *Ctest* we can find as an upper bound for the largest sequence of subformulas in the input formula ϕ. Hence $\#X \leq |\phi|$. As a result we have found an upper bound

$$4.|\phi|^2 + c_1.|\phi| + c_2$$

for the time needed to answer one call to *Ctest* as part of the calculation of $Ctest(,,\phi,)$, c_1 and c_2 being (implementation dependent) constants.

By combining the two upper bounds calculated above, we found the upper bound to the amount of time needed in the calculation of $Ctest(,,\phi,)$ as a whole to be of the order $4.|\phi|^3.2^{|\phi|}$.

As for the upper bound to the space needed in calculating $Ctest(,,\phi,)$, note that in the worst case a call to *Ctest* is replaced by two other calls plus a command to process the results. As the number of calls is $\sigma(\{;;\bullet\phi;\})$ and a call will take $2\#X \times |D|$ on the stack at the most (as each occurence of a subformula of ϕ will occur at most twice in the split sequent) an upper bound for the stack is $2.|\phi|^3.2^{|\phi|}$. We assume that to calculate a call to *Ctest* one needs to keep the initial sequences in the memory. As we also have to produce (at most) three new sequences and need some space for (at most) three formulas, a fair upper bound for the space needed in one call is $4.\#X \times |D| + 3.|D|$ or $4.|\phi|^2 + 3.|\phi|$. Hence the order of space needed to calculate $Ctest(,,\phi,)$ is $2.|\phi|^3.2^{|\phi|}$.

6.4 IpLtest: an IpL tester

The split sequents of *IpLtest* are of the form $K;M;N \circ S;T;U$ or $K;M;N \odot S;T;U$. As in the previous section, we will use the abbreviations L and R in describing the rules of *IpLtest*. Here $L = K;M;N$ and $R = S;T;U$. The notation $L \bullet R$ will be used to denote either $L \circ R$ or $L \odot R$.

Testing the derivability of formula A starts with the split sequent $;; \circ A;;$. Formulas to be treated are placed in N or S, those already treated are placed in K or U (and kept to facilitate the recognition of closure of a sequent). In *IpLtest* we have to take special care of implications and negations. Treatment on the righthand side (i.e. if implications or negations appear in S) is postponed; the implications and negations are placed in T. Formulas in T will only be treated if everything else fails.

On the lefthand side implications and negations may have to be treated more than once. After being treated, implications and negations are not moved from N to K, but to M. If $N = S = \emptyset$, then *IpLtest* may try all formulas in M again (by the RL-rule). To avoid *IpLtest* to go on with repeating the formulas in M indefinitely, there is a mechanism to keep track of the changes in the set of atoms on the lefthand side of the split sequent. We will write $L \circ R$ if there have not been introduced new atoms on the lefthand side since the last treatment of a formula in T. After the introduction of a 'new' atomic formula on the lefthand side, the split sequent is written as $L \odot R$. $L \odot R$ becomes $L' \circ R'$ via the RL-rule.

As in case of *CpLtest* a split sequent $L \bullet R$ is *closed* if $L \cap R \neq \emptyset$. As before we will assume that no *IpLtest* rules are applicable to a closed split sequent. Let p be an atomic formula. The rules of *IpLtest* are:

$$(pR)\ \frac{L \bullet p, R}{L \bullet R, p} \qquad\qquad (pL1)\ \frac{L, p \odot ; T; U}{p, L \odot ; T; U}$$

$$((pL2)\ \frac{L, p \circ ; T; U}{L \circ ; T; U} \quad p \in L \qquad\qquad (pL3)\ \frac{L, p \circ ; T; U}{p, L \odot ; T; U} \quad p \notin L$$

$$(\neg R)\ \frac{L \bullet \neg A, S; T; U}{L \bullet S; T, \neg A; U} \qquad\qquad (\neg L)\ \frac{K; M; N, \neg A \bullet ; T; U}{K; \neg A, M; N \bullet A; T; U}$$

$$(\wedge R)\ \frac{L \bullet A\wedge B, R}{L \bullet A, R, A\wedge B \quad L \bullet B, R, A\wedge B} \qquad (\wedge L)\ \frac{L, A\wedge B \bullet ; T; U}{A\wedge B, L, A, B \bullet ; T; U}$$

$$(\vee R)\ \frac{L \bullet A\vee B, R}{L \bullet A, B, R, A\vee B} \quad (\vee L)\ \frac{L, A\vee B \bullet ; T; U}{A\vee B, L, A \bullet ; T; U \quad A\vee B, L, B \bullet ; T; U}$$

$$(\to R)\ \frac{L \bullet A\to B, S; T; U}{L \bullet S; T, A\to B; U}$$

$$(\to L)\ \frac{K; M; N, A\to B \bullet ; T; U}{K; A\to B, M; N \bullet A; T; U \quad K; A\to B, M; N, B \bullet ; T; U}$$

$$\mathbf{RL}\ \frac{K; M; \odot ; T; U}{; K; M \circ ; T; U}$$

$$\neg\mathbf{RR}\ \frac{K; M; \circ ; \neg A, T; U}{K; M; A \circ ; ; \quad K; M; \circ ; T; U}$$

$$\to\mathbf{RR}\ \frac{K; M; \circ ; A\to B, T; U}{K; M; A \circ B; ; \quad K; M; \circ ; T; U}$$

IpLtest, like *CpLtest*, has an L- and an R-rule for each of the connectives. $\neg R$ and $\to R$ postpone the treatment of negations and implications until all other rules but $\neg$RR or $\to$RR have failed. The RR-rules are special in that they may decrease the number of formulas in the split sequent. The RL-rule enforces treatment of all implications and negations on the lefthand side of the split sequent. The RL-rule causes the sequence M (the implications and negations to be repeated) to make up the new sequence N (of formulas to be treated).

Note that for each split sequent at most one of the *IpLtest* rules is applicable. Hence the algorithm *IpLtest* is deterministic. To prove *IpLtest* to terminate on every split sequent we again define a measure of complexity on split sequents, as we did for *CpLtest*. This time however the definition is more complex.

6.4.0.1. Definition. *Let p be an atomic formula A an* **IpL** *formula, $L \bullet R$ a split sequent (such that $L = K; M; N$ and $R = S; T; U$) and X a set of split sequents.*

1. $\gamma(p) = 0$;
2. $\gamma(\neg A) = \gamma(A) + 2$;
3. $\gamma(A * B) = \gamma(A) + \gamma(B) + 3$ if $* \in \{\wedge, \vee\}$;
4. $\gamma(A\to B) = \gamma(A) + \gamma(B) + 4$;
5. $\delta(L \circ R) = \Sigma\{\gamma(A) + 2 \mid A \in N\} + \Sigma\{\gamma(A) + 2 \mid A \in S\}+$ $\Sigma\{\gamma(A) + 1 \mid A \in T\}$;
6. $\delta(L \odot R) = \delta(L \circ R) + 1$;
7. $\lambda(L \bullet R) = \Sigma\{\gamma(A) + 2 \mid (A = B\to C$ or $A = \neg B)$ and $A \in Sub(L \cup R)\}$;
8. $\eta(L \bullet R) = \#Sub(M \cup N) + \#Sub(S) + \#Sub(T)$;

9. $n(L \bullet R) = \#\{p \text{ atomic} \mid p \in Sub(L \cup R)\}$;
10. $m(L \bullet R) = \#\{p \text{ atomic} \mid p \in K\}$;
11. $\mu(L \bullet R) = (n(L \bullet R) - m(L \bullet R)).\lambda(L \bullet R) + \delta(L \bullet R)$;
12. $\sigma(X) = \Sigma\{2^{\eta(L \bullet R)} \times \mu(L \bullet R) \mid L \bullet R \in X\}$.

6.4.0.2. LEMMA. *If $L \bullet R$ a split sequent then $\mu(L \bullet R) \geq 0$. If $L \bullet R$ is derived from split sequent $L' \bullet R'$ by application of one of the* IpLtest *rules, then*

$$\mu(L \bullet R) < \mu(L' \bullet R')$$

If X a set of split sequents then $\sigma(X) \geq 0$. If X' a set of split sequents derived from X by application of one of the IpLtest *rules (replacing the split sequent treated by the result(s) of the application of the* IpLtest *rule) then*

$$\sigma(X') < \sigma(X)$$

Proof. By checking the rules. In most cases application of a rule wille decrease the δ of the split sequent. Only the RL-rule increases the δ. However, for a given split sequent the $pL3$-rule can only be applied $n - m$-times (as n is the total number of atoms in the split sequent and m the number of atoms in K). Hence, also the RL-rule can only be applied $n - m$ times. The number λ, as defined above, is an upper bound on the increase of δ by an application of the RL-rule. ⊣

6.4.0.3. DEFINITION. *A split sequent $L \bullet R$ is closing ($L \underline{\bullet} R$) if*

1. *$L \bullet R$ is closed (i.e. $L \cap R \neq \emptyset$);*
2. *one of the RR-rules is applicable and one of the resulting split sequents is closing;*
3. *one of the other rules is applicable and its resulting split sequent(s) is (are) closing.*

To prove *IpLtest* to be sound and complete we will prove

$$L \underline{\bullet} R \quad \Leftrightarrow \quad L \vdash \bigvee R$$

In order to do so, we need the following definition and some facts.

6.4.0.4. DEFINITION. *A split sequent $L \bullet R$ is* reduced *if it is not closed and no other rules but the RR-rules are applicable. If $L \bullet R$ a split sequent that is not closing, application of the* IpLtest *rules, with the exception of the RR-rules, will result in one or more reduced split sequents that will be called* reductions *of $L \bullet R$.*

Note that a split sequent is reduced if not closed and $N = S = \emptyset$.

6.4.0.5. FACT. *If $L \bullet R$ is a reduced split sequent then:*

1. $A \wedge B \in L \quad \Rightarrow \quad A \in L$ *and* $B \in L$;
2. $A \vee B \in L \quad \Rightarrow \quad A \in L$ *or* $B \in L$;
3. $\neg A \in L \quad \Rightarrow \quad A \notin L$;
4. $A \rightarrow B \in L \quad \Rightarrow \quad A \notin L$ *or* $B \in L$;
5. $A \wedge B \in R \quad \Rightarrow \quad A \in R$ *or* $B \in R$;
6. $A \vee B \in R \quad \Rightarrow \quad A \in R$ *and* $B \in R$.

The truth of this fact can be established by observation of the *IpLtest* rules. No rule changes the monotone increase of the set of formulas L and only with the RR-rules do formulas disappear from R. Note that a reduced split sequent is always of the form $K; M; \circ; T; U$.

6.4.0.6. LEMMA. *$L \mathrel{\underline{\bullet}} R$ implies $L \vdash \bigvee R$ (in* **IpL***).*

Proof. For a closed split sequent the lemma is obvious. As the tableau for a split sequent is a finite tree of split sequents, we can proceed by induction on the depth of the sequent (closed split sequents having depth zero).

According to the *IpLtest* rules $L \mathrel{\underline{\bullet}} A \wedge B, R$ iff both $L \mathrel{\underline{\bullet}} A, R, A \wedge B$ and $L \mathrel{\underline{\bullet}} B, R, A \wedge B$. By induction hypothesis we may infer $L \vdash A \vee (A \wedge B) \vee \bigvee R$ and $L \vdash B \vee (A \wedge B) \vee \bigvee R$. Hence in **IpL** one can derive $L \vdash (A \wedge B) \vee \bigvee R$.

All *IpLtest* rules can be treated in the same way. For the RR-rules observe that only one of the consequents of the rules has to be closing.

For the *RR*$\rightarrow$-rule for example: if $L, A \mathrel{\underline{\bullet}} B$ then by induction hypothesis $L, A \vdash B$ and hence $L \vdash A \rightarrow B$. Otherwise if $L \mathrel{\underline{\bullet}} R$ and hence $L \vdash \bigvee R$, then of course also $L \vdash A \rightarrow B \vee \bigvee R$. For the *RR*$\neg$-rule, in case $R = \emptyset$, note that $\bigvee \emptyset = \bot$. $\dashv$

To prove $L \bullet R$ is not closing implies $L \nvdash \bigvee R$, we will extract from the non-closing tableau a Kripke model K forcing all formulas in L and none of those in R. In the definition of the Kripke model we will make use of the concept of the leftmost non-closing reduction of a split sequent. In finding this reduction one chooses to follow the leftmost non-closing conclusion of each *IpLtest* rule.

6.4.0.7. DEFINITION. *Let $L \bullet R$ be a non-closing split sequent. The Kripke model K* associated with *$L \bullet R$ is defined as the ordered set of (leftmost) non-closing reduced split sequents:*

1. *the leftmost non-closing reduction of $L \bullet R$ is the root of K;*
2. *if $k_l \in K$ corresponds to the split sequent $L' \circ; A \rightarrow B, T; U$ and $T \neq \emptyset$, then the leftmost non-closing reductions of $L', A \circ B;;$ and $L' \circ; T; U$ are nodes of K, say respectively k_m and k_n, such that $k_l \leq k_m$ and $k_l \leq k_n$;*
3. *if $k_l \in K$ corresponds to the split sequent $L' \circ; A \rightarrow B, T; U$ and $T = \emptyset$, then the leftmost non-closing reduction of $L', A \circ B;;$ is a node of K, say k_m, such that $k_l \leq k_m$;*
4. *if $k_l \in K$ corresponds to the split sequent $L' \circ; \neg A, T; U$ and $T \neq \emptyset$, then the leftmost non-closing reductions of $L', A \circ;;$ and $L' \circ; T; U$ are nodes of K, say respectively k_m and k_n, such that $k_l \leq k_m$ and $k_l \leq k_n$;*
5. *if $k_l \in K$ corresponds to the split sequent $L' \circ; \neg A, T; U$ and $T = \emptyset$ then the leftmost non-closing reduction of $L', A \circ;;$ is a node of K, say k_m, such that $k_l \leq k_m$;*
6. *the order relation $\leq$ is reflexive and transitive;*
7. *if $k_l \in K$ is the node corresponding to $L' \circ R'$, then $k_l \Vdash p$ for atomic formulas p iff $p \in L$.*

6.4.0.8. LEMMA. *If $L \odot R$ is a non-closing split sequent and K its associated Kripke model, with root k_0, then for each formula A we have $A \in L \quad \Rightarrow \quad k_0 \Vdash A$ and $A \in R \quad \Rightarrow \quad k_0 \nVdash A$.*

Proof. First observe that if $L' \circ R'$ is the leftmost non-closing reduction of $L \bullet R$, and for each formula A we would have $A \in L' \quad \Rightarrow \quad k_0 \Vdash A$ and $A \in R' \quad \Rightarrow \quad k_0 \nVdash A$, then the lemma is a consequence of the *IpLtest* rules (all except the RR-rules are reversible).

With induction on the length of formula A we will prove that if $k_l \in K$ corresponds to the reduced split sequent $L' \circ R'$, then $A \in L'$ implies $k_l \Vdash A$ and $A \in R'$ implies $k_l \nVdash A$.

The cases where A is atomic, a conjunction or a disjunction are obvious (using fact 6.4.0.5).

Let $A = B \to C$ and $A \in L'$. Let $k_l \leq k_m$ and $k_m \in K$ correspond to a reduced split sequent $L'' \circ R''$ and $k_m \Vdash B$. As formula A has been treated in the derivation of $L'' \circ R''$, there is a $k_n \in K$, $k_m \leq k_n$ and $k_n \nVdash B$ or $k_n \Vdash C$ such that k_m and k_n force the same atoms. This is due to the fact that the RL-rule would have been applied between the sequents of k_n and k_m if there was a difference in the atoms forced. By a simple lemma on Kripke models k_m and k_n force the same formulas and hence $k_m \Vdash C$, which proves $k_l \Vdash B \to C$.

Let $A = B \to C$ and $A \in R'$. Note that $A \in T$ and the split sequent $L', B \circ C; ;$ (appearing after one or more applications of an RR-rule) will not be closing. Hence, if k_m corresponds to the leftmost non-closing reduction of $L', B \circ C; ;$, by the induction hypothesis $k_m \nVdash A$. As we have $k_l \leq k_m$ we infer that $k_l \nVdash A$. $\dashv$

6.4.0.9. THEOREM. *A split sequent $L \odot R$ is closing, using the* IpLtest *rules, iff $L \vdash \bigvee R$.*

Proof. By combining the previous two lemmas. $\dashv$

The following pseudo-code program, *Itest* is an implementation of the *IpLtest* algorithm. For the language conventions see the *Ctest* program in the previous section. To test whether a formula ϕ is a theorem of **IpL**, one calls the program $Itest(,,,\phi,,,\mathbf{false})$, where the input corresponds to the initial sequent $;; \circ \phi;;$ for *IpLtest*.

```
Itest(K, M, N, S, T, U : sequence of formula, d : bool) : bool
  if S ≠ ∅
  then let S = ⟨A, S'⟩
         if A ∈ K ∪ M ∪ N then true
         else in case A
              atomic : Itest(K, M, N, S', T, ⟨A, U⟩, d)
              ¬B     : Itest(K, M, N, S', ⟨A, T⟩, U, d)
              B ∧ C  : if Itest(K, M, N, ⟨B, S'⟩, T, ⟨A, U⟩, d)
                       then Itest(K, M, N, ⟨C, S'⟩, T, ⟨A, U⟩, d)
                       else false
              B ∨ C  : Itest(K, M, N, ⟨B, C, S'⟩, T, ⟨A, U⟩, d)
              B→C    : Itest(K, M, N, S', ⟨A, T⟩, U, d)
  else if N ≠ ∅
         then let N = ⟨A, N'⟩
              if A ∈ T ∪ U then true
              else in case A
                   atomic : if A ∉ K
                            then Itest(⟨A, K⟩, M, N', , T, U, true)
                            else Itest(K, M, N', , T, U, d)
                   ¬B     : Itest(K, ⟨A, M⟩, N', B, T, U, d)
                   B ∧ C  : Itest(⟨A, K⟩, M, ⟨A, B, N'⟩, , T, U, d)
                   B ∨ C  : if Itest(⟨A, K⟩, M, ⟨B, N'⟩, , T, U, d)
                            then Itest(⟨A, K⟩, M⟨C, N'⟩, , T, U, d)
                            else false
                   B→C    : if Itest(K, ⟨A, M⟩, N', B, T, U, d)
                            then Itest(K, ⟨A, M⟩, ⟨C, N'⟩, , T, U, d)
                            else false
         else if d then Itest(K, , M, , T, U, false)
              else if T ≠ ∅
                   then let T = ⟨A, T'⟩
                             in case A
                             ¬B     : if Itest(K, M, B, , , , d) then true
                                      else Itest(K, M, , , T', U, d)
                             B→C    : if Itest(K, M, B, C, , , d) then true
                                      else Itest(K, M, , , T', U, d)
                   else false
```

6.5 Ktest: a tester for K

In this section and the following we will introduce tableaux testers for modal propositional logic.

The first tester to be described is *Ktest*, a tester for the modal logic **K**, that will act as the minimal system for the modal logics in this section. The axioms of

K are those of classical propositional logic **CpL** plus $\Box(A{\to}B){\to}(\Box A{\to}\Box B)$ and necessitation ($\vdash A \quad\Rightarrow\quad \vdash \Box A$) as an extra derivation rule.

Split sequents of *Ktest* are of the form $K; M; N \bullet S; T; U$. The rules for *Ktest* are the rules of *CpLtest*, with rules added to deal with $\Box$ and $\Diamond$:

$$(pR)\ \frac{L \ \bullet\ p, R}{L \ \bullet\ R, p} \qquad\qquad (pL)\ \frac{L, p \bullet ; T; U}{p, L \bullet ; T; U}$$

$$(\neg R)\ \frac{L \ \bullet\ \neg A, R}{L, A \ \bullet\ R, \neg A} \qquad\qquad (\neg L)\ \frac{L, \neg A \bullet ; T; U}{\neg A, L \ \bullet\ A; T; U}$$

$$(\wedge R)\ \frac{L \ \bullet\ A \wedge B, R}{L \ \bullet\ A, R, A \wedge B \quad L \ \bullet\ B, R, A \wedge B} \qquad (\wedge L)\ \frac{L, A \wedge B \bullet ; T; U}{A \wedge B, L, A, B \bullet ; T; U}$$

$$(\vee R)\ \frac{L \ \bullet\ A \vee B, R}{L \ \bullet\ A, B, R, A \vee B} \qquad (\vee L)\ \frac{L, A \vee B \bullet ; T; U}{A \vee B, L, A \bullet ; T \quad A \vee B, L, B \bullet ; T; U}$$

$$(\to R)\ \frac{L \ \bullet\ A{\to}B, R}{L, A \ \bullet\ B, R, A{\to}B} \qquad (\to L)\ \frac{L, A{\to}B \bullet ; T; U}{A{\to}B, L \ \bullet\ A; T \quad A{\to}B, L, B \bullet ; T; U}$$

$$(\Box R)\ \frac{L \ \bullet\ \Box A, S; T; U}{L \ \bullet\ S; T, \Box A; U} \qquad\qquad (\Box L)\ \frac{K; M; N, \Box A \bullet ; T; U}{K; \Box A, M; N \bullet ; T; U}$$

$$(\Diamond R)\ \frac{K; M; N \ \bullet\ \Diamond A, S; T; U}{K; \Box\neg A, M; N \ \bullet\ S; T; U, \Diamond A} \qquad\qquad (\Diamond L)\ \frac{L, \Diamond A \bullet ; T; U}{\Diamond A, L \bullet ; T, \Box\neg A; U}$$

The NW-rule

$$\frac{K; M; \ \bullet ; \Box A, T; U}{; ; M^* \ \bullet\ A; ; \quad K; M; \ \bullet ; T; U} \text{ where } M^* = \{B \mid \Box B \in M\}$$

The NW-rule (the *new world rule*) plays the same role as the RR-rules in *IpLtest*. If the algorithm is regarded as a method of systematically constructing a Kripke model (that is a counterexample to the formula tested and failure of which proves that it is a theorem is true) this rule forces the introduction of a new world in the model construction.

The rules in *Ktest* for the possibility operator differ from the other rules (in *Ktest*, *IpLtest* or *CpLtest*), as in treating the formula $\Diamond A$, we not only use the subformula A, but in the result of the $\Diamond$-rules there appears a formula $\Box\neg A$. This is best understood as treating $\Diamond$ as an abbreviation of $\neg\Box\neg$. In this way we somewhat restricted the number of rules. From the rules above it can be proved that we get an equivalent system by adding a new $\Diamond$NW-rule:

$$\frac{K; M, \Diamond A;\ \bullet\ ;\,;U}{;\,; M^*, A \bullet ;\,; \quad K; M;\ \bullet\ ;\,;U} \text{ where } M^* = \{B \mid \Box B \in M\}$$

and replacing the $\Diamond$ rules above by:

$$(\Diamond R)\ \frac{L\ \bullet\ \Diamond A, S; T; U}{L\ \bullet\ S; T, \Diamond A; U} \qquad (\Diamond L)\ \frac{K; M; N, \Diamond A \bullet ; T; U}{K; \Diamond A, M; N \bullet ; T; U}$$

Define a *Ktest* split sequent $L\ \bullet\ R$ to be *closed* if $L \cap R \neq \emptyset$. None of the *Ktest* rules is applicable to a closed split sequent.

The rules of *Ktest* are named according to the kind of formula treated and its position. Hence we have a pL- and a pR-rule, an $\rightarrow\!L$- and an $\rightarrow\!R$-rule and so on.

Note that at most one rule is applicable to any split sequent and hence the algorithm *Ktest* is deterministic. To prove the algorithm *Ktest* to terminate on each split sequent, we define a measure of complexity on a set X of split sequents, $\sigma(X)$ that will strictly decrease with each application of a *Ktest* rule on a member of X. Application of a *Ktest* rule to X has as its result a new set of split sequents X', where the split sequent treated in X is replaced by the result from the application of the *Ktest* rule.

6.5.0.1. DEFINITION. *Let p be an atomic formula, A a* **K** *formula $L\ \bullet\ R$ a split sequent (such that $L = M; N$ and $R = S; T$) and X a set of split sequents.*

1. $\gamma(p) = 0$;
2. $\gamma(\neg A) = \gamma(A) + 1$;
3. $\gamma(A \circ B) = \gamma(A) + \gamma(B) + 2$ if $\circ \in \{\wedge, \vee, \rightarrow\}$;
4. $\gamma(\Box A) = \gamma(A) + 1$;
5. $\gamma(\Diamond A) = \gamma(A) + 3$;
6. $\mu(L\ \bullet\ R) = \Sigma\{\gamma(A) + 1 \mid A \in N\} + \Sigma\{\gamma(A) \mid A \in M\} +$ $\Sigma\{\gamma(A) + 1 \mid A \in S\} + \Sigma\{\gamma(A) \mid A \in T\}$;
7. $\eta(L\ \bullet\ R) = \#Sub(M) + \#Sub(N) + \#Sub(S) + \#Sub(T)$;
8. $\sigma(X) = \Sigma\{2^{\eta(L \bullet R)} \times \mu(L \bullet R) \mid L \bullet R \in X\}$.

6.5.0.2. LEMMA. *If $L\ \bullet\ R$ a split sequent then $\mu(L\ \bullet\ R) \geq 0$. If $L\ \bullet\ R$ is a split sequent derived from split sequent $L'\ \bullet\ R'$ by application of one of the* Ktest *rules, then*

$$\mu(L\ \bullet\ R) < \mu(L'\ \bullet\ R')$$

If X a set of split sequents then $\sigma(X) \geq 0$. If X' a set of split sequents derived from X by application of one of the Ktest *rules (replacing the split sequent treated by the result(s) of the application of the* Ktest *rule) then*

$$\sigma(X') < \sigma(X)$$

Proof. By simply checking the rules. $\dashv$

6.5.0.3. DEFINITION. *A split sequent $L \bullet R$ is closing ($L \,\underline{\bullet}\, R$) if*

1. *$L \bullet R$ is closed;*
2. *the NW-rule is applicable and one of the resulting split sequents is closing;*
3. *the $\wedge$R-, $\vee$L- or $\rightarrow$L-rule is applicable and both the resulting split sequents are closing;*
4. *one of the other rules is applicable and its resulting split sequent is closing.*

To prove *Ktest* to be sound and complete we will prove

$$L \,\underline{\bullet}\, R \quad \Leftrightarrow \quad L \vdash \bigvee R$$

But to do so we need the following definition and facts.

6.5.0.4. DEFINITION. *A split sequent $L \bullet R$ is* reduced *if it is not closed and $N = S = \emptyset$ (no other rules but the NW-rule are applicable). If $L \bullet R$ is a split sequent that is not closing, application of the* Ktest *rules, with the exception of the NW-rule, will result in one or more reduced split sequents that will be called* reductions *of $L \bullet R$.*

A fortiori a split sequent is reduced if it is not closed and no *Ktest* rule applies to it.

6.5.0.5. FACT. *If $L \bullet R$ is a reduced split sequent then:*

1. $A \wedge B \in L \quad \Rightarrow \quad A \in L$ *and* $B \in L$;
2. $A \vee B \in L \quad \Rightarrow \quad A \in L$ *or* $B \in L$;
3. $\neg A \in L \quad \Rightarrow \quad A \in R$;
4. $A \rightarrow B \in L \quad \Rightarrow \quad A \in R$ *or* $B \in L$;
5. $A \wedge B \in R \quad \Rightarrow \quad A \in R$ *or* $B \in R$;
6. $A \vee B \in R \quad \Rightarrow \quad A \in R$ *and* $B \in R$;
7. $\neg A \in R \quad \Rightarrow \quad A \in L$;
8. $A \rightarrow B \in R \quad \Rightarrow \quad A \in L$ *and* $B \in R$.

The truth of this fact can be established by observation of the *Ktest* rules. Observe that only the NW-rule changes the monotonic increase of the sets of formulas L and R.

6.5.0.6. LEMMA. *If a split sequent $L \bullet R$ is closing (by the* Ktest *rules) then $L \vdash \bigvee R$ (in* **K***).*

Proof. For a closed split sequent the lemma is obvious. As the tableau for a split sequent is a finite tree of split sequents, we can proceed by induction on the depth of the sequent (closed split sequents having depth zero).

According to the *Ktest* rules $L \,\underline{\bullet}\, A \wedge B, R$ iff both $L \,\underline{\bullet}\, A, R, A \wedge B$ and $L \,\underline{\bullet}\, B, R, A \wedge B$. By the induction hypothesis we may infer $L \vdash A \vee A \wedge B \vee \bigvee R$ and $L \vdash B \vee A \wedge B \vee \bigvee R$. Hence in **K** one can derive $L \vdash A \wedge B \vee \bigvee R$.

All *Ktest* rules can be treated in the same way. For the NW-rule observe that only one of the consequents of the rule has to be closing.

As M contains only boxed formulas, from $M^* \vdash A$ infer (by necessitation) that $M \vdash \Box A$. Hence we may conclude that $K, M \vdash \Box A \vee \bigvee T \vee \bigvee U$. If on the other hand it should be the case that $K, M \vdash \bigvee T \vee \bigvee U$ then obviously we would have $K, M \vdash \Box A \vee \bigvee T \vee \bigvee U$. $\dashv$

To prove that if $L \bullet R$ is not closing, then $L \nvdash \bigvee R$ we will extract from the non-closing tableau a Kripke model K forcing all formulas in L and none of those in R. In the definition of the Kripke model we will make use of the concept of the leftmost non-closing reduction of a split sequent as we did for *IpLtest*. In finding this reduction one chooses to follow the leftmost non-closing conclusion of each *Ktest* rule for which there is a choice.

6.5.0.7. Definition. *Let $L \bullet R$ be a non-closing split sequent. The Kripke model K associated with $L \bullet R$ is defined as a set of (leftmost) non-closing reduced split sequents, with an irreflexive relation $<$:*

1. *the leftmost non-closing reduction of $L \bullet R$ is the root of K;*
2. *if $k_l \in K$ corresponds to the split sequent $K; M; \bullet; T; U$ and $T = \{\Box A_1, \ldots \Box A_t\} \neq \emptyset$ then the leftmost non-closing reductions of $;; M^* \bullet A_i;;$ (where $M^* = \{B \mid \Box B \in M\}$ and $\Box A_i \in T$) are nodes of K, say respectively $l_1, \ldots, l_t$, such that for all i such that $1 \le i \le t$: $k_l < l_i$;*
3. *if $k_l \in K$ corresponds to the split sequent $K; M; \bullet;; U$ (hence $T = \emptyset$) then k_l is a terminal node in K.*
4. *if $k_l \in K$ is the node corresponding to $L' \bullet R'$, then $k_l \Vdash p$ for atomic formulas p iff $p \in L$.*

6.5.0.8. Lemma. *If $L \bullet R$ is a non-closing split sequent and K its associated Kripke model, with root k_0, then for each formula A we have $A \in L \quad \Rightarrow \quad k_0 \Vdash A$ and $A \in R \quad \Rightarrow \quad k_0 \nVdash A$.*

Proof. First observe that if $L' \bullet R'$ is the leftmost non-closing reduction of $L \bullet R$, and for each formula A we would have $A \in L' \quad \Rightarrow \quad k_0 \Vdash A$ and $A \in R' \quad \Rightarrow \quad k_0 \nVdash A$, then the lemma is a consequence of the *Ktest* rules (all except the NW-rule are reversible).

With induction on the length of formula A we will prove that if $k_l \in K$ corresponds to the reduced split sequent $L' \bullet R'$, then $A \in L'$ implies $k_l \Vdash A$ and $A \in R'$ implies $k_l \nVdash A$.

The cases where A is atomic, a conjunction, a disjunction or an implication are obvious (using fact 6.5.0.5).

Let $A = \Box B$ and $A \in L'$. As $L' \bullet R'$ is reduced A will be a member of M. Let $k_l < k_m$ and $k_m \in K$ correspond to a reduced split sequent $L'' \bullet R''$. Then $L'' \bullet R''$ will be a result of the application of the NW-rule and hence we will have $B \in L''$. By induction hypothesis conclude that $k_m \Vdash B$. Which proves that $k_l \Vdash \Box B$. Note that if k_l is a terminal node then, as K has been defined in such a way that it is irreflexive, trivially $k_l \Vdash \Box B$.

If $A = \Box B$ and $A \in R'$ then A will be in T and application of the NW-rule (and reduction) will result in a node k_m corresponding to a reduced split sequent $L'' \bullet R''$

such that $k_l < k_m$ and $B \in R''$. Using the induction hypothesis we conclude $k_m \nVdash B$ and hence $k_l \nVdash \Box B$. $\dashv$

6.5.0.9. THEOREM. *A split sequent $L \bullet R$ is closing, using the* Ktest *rules, iff $L \vdash \bigvee R$.*

Proof. By combining the previous two lemmas. $\dashv$

Like we did previously for *CpLtest* and *IpLtest*, we will give a pseudo-code translation of the algorithm *KMtest*.

```
KMtest(K, M, N, S, T, U : sequence of formula) : bool
   if S ≠ ∅
   then let S = ⟨A, S'⟩
          if A ∈ J ∪ K ∪ M ∪ N then true
          else in case A
                atomic : KMtest(K, M, N, S', T, ⟨A, U⟩)
                ¬B     : KMtest(K, M, ⟨B, N⟩, S', T, U)
                B ∧ C  : if KMtest(K, M, N, ⟨B, S'⟩, T, ⟨A, U⟩)
                         then KMtest(K, M, N, ⟨C, S'⟩, T, ⟨A, U⟩)
                         else false
                B ∨ C  : KMtest(K, M, N, ⟨B, C, S'⟩, T, ⟨A, U⟩)
                B→C    : KMtest(K, M, ⟨B, N⟩, ⟨C, S'⟩, T, ⟨A, U⟩)
                □B     : KMtest(K, M, N, S', ⟨A, T⟩, U)
                ◇B     : KMtest(K, M, N, S', ⟨¬B, T⟩, U)
   else if N ≠ ∅
          then let N = ⟨A, N'⟩
                if A ∈ T ∪ U then true
                else in case A
                      atomic KMtest(⟨A, K⟩, M, N', , T, U)
                      ¬B     : KMtest(⟨A, K⟩, M, N', B, T, U)
                      B ∧ C  : KMtest(⟨A, K⟩, M, ⟨A, B, N'⟩, , T, U)
                      B ∨ C  : if KMtest(⟨A, K⟩, M, ⟨B, N⟩, , T, U)
                               then KMtest(⟨A, K⟩, M⟨C, N⟩, , T, U)
                               else false
                      B→C    : if KMtest(⟨A, K⟩, M, N, B, T, U)
                               then KMtest(⟨A, K⟩, M, ⟨C, N⟩, , T, U)
                               else false
                      □B     : KMtest(K, ⟨A, M⟩, N, , T, U)
                      ◇B     : KMtest(K, M, N, , ⟨T, A⟩, U)
          else if T ≠ ∅
                then let T = ⟨□A, T'⟩ and M* = {B | □B ∈ M}
                       if KMtest(, , M*, A, , ) then true
                       else KMtest(K, M, , , T, U)
                else false
```

6.6 Other testers for modal propositional logic

Testers for other modal propositional logics can be derived from *Ktest* by changing some of the rules (mainly the NW-rule). In this section we will indicate for several modal logics how a tester algorithm may be obtained.

6.6.1 Ttest: a T tester

The modal logic **T** has as its axioms and rules those of **K** plus the axiom $\Box\phi\to\phi$. **T** is complete for finite reflexive Kripke models (a proof can be found in [HC 84]).

To obtain *Ttest*, a tester for the modal logic **T**, we only have to change the $\Box$L-rule in *Ktest*.

6.6.1.1. DEFINITION. *The tester* Ttest *has the same rules as* Ktest, *except for the $\Box L$-rule that is replaced by:*

$$(\mathbf{T}\Box L)\ \frac{K; M; N, \Box A \bullet ; T; U}{K; \Box A, M; N, A \bullet ; T; U}.$$

6.6.1.2. THEOREM. *A split sequent $L \bullet R$ is closing, using the* Ttest *rules, iff $L \vdash \bigvee R$.*

Proof. The proof is essentially as for theorem 6.5.0.9, using amended versions of lemma 6.5.0.6 and lemma 6.5.0.8.

As for lemma 6.5.0.6, note that in **T**, using $\Box A \vdash A$, from $L, \Box A, A \vdash \bigvee R$ we may infer $L, \Box A \vdash \bigvee R$.

To prove an amended version of lemma 6.5.0.8, we have to change the definition of an associated Kripke model, definition 6.5.0.7, in such a way that the resulting model is always reflexive. Note that the change in the $\Box L$-rule reflects the axiom $\Box\phi\to\phi$ of **T**. If $L \bullet R$ is a split sequent and $\Box A \in L$, then in the leftmost non-closing reduction of $L \bullet R$, by the $\Box L$-rule, we will have $A \in L$. This is exactly what we need to change the proof of lemma 6.5.0.8 to apply to **T**. $\dashv$

6.6.2 K4test: a K4 tester

The modal logic **K4** has as its axioms and rules those of **K** plus the axiom $\Box\phi\to\Box\Box\phi$. A proof that **K4** is complete for finite transitive Kripke models can be found in [HC 84]. For the definition of *K4test*, a tester for the modal logic **K4**, we will extend the split sequents of *Ktest*. A split sequent of *K4test* is of the form $K; M; N\ (w; W)\ S; T; U$, where $K; M; N \bullet S; T; U$ is a split sequent of *Ktest*, w a *world*, a tuple $\langle X, Y\rangle$, with X and Y sets of formulas, and W a sequence of worlds.

The algorithm of *K4test*, given below, is obtained by amending the rules of *Ktest* for the split sequents of *K4test*, changing the NW-rule and adding a new rule restricting the applicability of the *K4test* rules.

The K4NW-rule is

$$\frac{K;M;\ (w,W)\ ;\Box A,T;U}{;M;M^*\ (w';W,w)\ A;;\quad K;M;\ (w;W)\ ;T;U}$$

where $M^* = \{B \mid \Box B \in M\}$.

This rule reflects the transitivity of the frames where the axiom $\Box\phi{\to}\Box\Box\phi$ is valid, by repeating all boxed formulas that have appeared on the left-hand side of the reduced split sequent.

The new rule of non-applicability declares that for a sequent $L(w;W)R$ with $w \in W$ no rule of *K4test* is applicable. In particular this may be the result of the K4NW-rule, if the world $w' = \langle M \cup M^*, \{A\}\rangle$ already occures in the list W, w of worlds that appeared above this split sequents in its construction from the starting split sequents, using the *K4test*-rules.

For the rules of *K4test* we use the same conventions as for *Ktest* and we will abbreviate $K;M;N\ (w;W)\ S;T;U$ by $L\ (w;W)\ R$.

$$(pR)\ \frac{L\ (w;W)\ p,R}{L\ (w;W)\ R,p} \qquad (pL)\ \frac{L,p\ (w;W)\ ;T;U}{p,L\ (w;W)\ ;T;U}$$

$$(\neg R)\ \frac{L\ (w;W)\ \neg A,R}{L,A\ (w;W)\ R,\neg A} \qquad (\neg L)\ \frac{L,\neg A\ (w;W)\ ;T;U}{\neg A,L\ (w;W)\ A;T;U}$$

$$(\wedge R)\ \frac{L\ (w;W)\ A\wedge B,R}{L\ (w;W)\ A,R,A\wedge B\quad L\ (w;W)\ B,R,A\wedge B}$$

$$(\wedge L)\ \frac{L,A\wedge B\ (w;W)\ ;T;U}{A\wedge B,L,A,B\ (w;W)\ ;T;U} \qquad (\vee R)\ \frac{L\ (w;W)\ A\vee B,R}{L\ (w;W)\ A,B,R,A\vee B}$$

$$(\vee L)\ \frac{L,A\vee B\ (w;W)\ ;T;U}{A\vee B,L,A\ (w;W)\ ;T\quad A\vee B,L,B\ (w;W)\ ;T;U}$$

$$(\to R)\ \frac{L\ (w;W)\ A{\to}B,R}{L,A\ (w;W)\ B,R,A{\to}B}$$

$$(\to L)\ \frac{L,A{\to}B\ (w;W)\ ;T;U}{A{\to}B,L\ (w;W)\ A;T\quad A{\to}B,L,B\ (w;W)\ ;T;U}$$

$$(\Box R)\ \frac{L\ (w;W)\ \Box A,S;T;U}{L\ (w;W)\ S;T,\Box A;U} \qquad (\Box L)\ \frac{K;M;N,\Box A\ (w;W)\ ;T;U}{K;\Box A,M;N\ (w;W)\ ;T;U}$$

$$(\Diamond R)\ \frac{K;M;N\ (w;W)\ \Diamond A,S;T;U}{K;\Box\neg A,M;N\ (w;W)\ S;T,\Box A;U} \qquad (\Diamond L)\ \frac{L,\Diamond A\ (w;W)\ ;T;U}{L\ (w;W)\ ;T,\Box\neg A;U}$$

The K4NW-rule

$$\frac{K; M;\ (w; W)\ ; \Box A, T; U}{; M; M^*\ (\langle M \cup M^*, \{A\}\rangle; W, w)\ A; ; \quad K; M;\ (w; W)\ ; T; U}$$

where $M^* = \{B \mid \Box B \in M\}$.

Note that the top sequent of the K4NW-rule will be called closing if one of the resulting split sequents is closing.

6.6.2.3. LEMMA. *The algorithm* K4test *is deterministic and terminates on the input of any split sequent.*

Proof. To see that *K4test* is deterministic, it can be verified that for each split sequent at most one rule of *K4test* is applicable.

To prove that *K4test* terminates on every split sequent, we can define a measure of complexity, $\sigma(X)$, on a set X of split sequents, like we did for *Ktest* in definition 6.5.0.1. We will not spell out this definition here, but the only difference with the one for *Ktest* will be a contribution for the $(w; W)$ part in the split sequent.

Let the initial sequent be $L(w; W)R$. The worlds that may appear in the application of the *K4test* rules to this sequent (and its resulting sequents) are tuples $\langle M, A\rangle$, where $M \cup A$ is a set of subformulas in the initial sequent $L(w; W)R$. If m is the number of these world-like tuples that may be made out of $L(w; W)R$ and n the number of worlds in W in the initial sequent, then for every split sequent $L'(w'; W')R'$ that may be developed out of $L(w, W)R$ we have measure

$$v(L'(w', W')R') = m + n - \#W' + 1$$

that is strictly decreasing after each non-closing application of the K4NW-rule.

Taking this v into account, one can construct a strictly decreasing measure of complexity on a set of split sequents, as in definition 6.5.0.1. $\dashv$

To prove the counterpart of theorem 6.5.0.9 for *K4test*, we proceed as in section 6.5.

6.6.2.4. LEMMA. *If a split sequent $L(w; W)R$ is closing (by the* K4test *rules) then $L \vdash \bigvee R$ (in* **K4***).*

Proof. For a closed split sequent the lemma is obvious. As the tableau for a split sequent is a finite tree of split sequents, we can proceed by induction on the depth of the sequent (closed split sequents having depth zero).

All *K4test* rules can be treated as in the proof of lemma 6.5.0.6, except for the rule K4NW.

If $K; M;\ (w; W)\ ; T; U$ is closing then, by the induction hypothesis, $K, M \vdash \bigvee T \vee \bigvee U$. Then obviously also $K, M \vdash \Box A \vee \bigvee T \vee \bigvee U$.

On the other hand, if $; M; M^*\ (\langle M \cup M^*, \{A\}\rangle; W, w)\ A; ;$ is closing, then, by the induction hypothesis, $M, M^* \vdash A$. Applying the necessitation rule, infer that $M \vdash \Box A$, as $M = \{\Box B \mid B \in M^*\}$ and by the **K4** axiom $M \vdash \Box \bigwedge M$. $\dashv$

For the proof of the following lemma, we will, as in section 6.5, associate a Kripke model to a non-closing split sequent $L(w; W)R$.

6.6.2.5. DEFINITION. *Let $L(w;W)R$ be a non-closing split sequent.* The Kripke model K associated with $L(w;W)R$ *is defined as a set of (leftmost) non-closing reduced split sequents, with a transitive relation ρ:*

1. *the leftmost non-closing reduction of $L(w;W)R$ is the root of K;*
2. *if $k_l \in K$ corresponds to the split sequent $K;M;(w;W);T;U$ and $T = \{\Box A_1, \ldots \Box A_t\} \neq \emptyset$ then the leftmost non-closing reductions of $;M;M^* \bullet A_i;;$ (where $M^* = \{B \mid \Box B \in M\}$ and $\Box A_i \in T$) are nodes of K, say respectively $l_1, \ldots, l_t$, such that for all i such that $1 \leq i \leq t$: $k_l \rho l_i$;*
3. *if $L(w;W)R$ is non-closing because of $w \in W$ and w was introduced in W by application of the K4NW-rule to $L'(w;W')R'$, then, if $L'(w;W')R$ corresponds to k_l and $L(w;W)R$ corresponds to k_m, we identify k_l and k_m.*
4. *if $k_l \in K$ corresponds to the split sequent $K;M;(w;W);;U$ (hence $T = \emptyset$) then k_l is a terminal node in K.*
5. *if $k_l \in K$ is the node corresponding to $L'(w;W)R'$, then $k_l \Vdash p$ for atomic formulas p iff $p \in L$.*

6.6.2.6. LEMMA. *If $L(w;W)R$ is a non-closing split sequent and K its associated Kripke model, with root k_0, then for each formula A we have $A \in L \quad \Rightarrow \quad k_0 \Vdash A$ and $A \in R \quad \Rightarrow \quad k_0 \nVdash A$.*

Proof. First observe that if $L'(w;W)R'$ is the leftmost non-closing reduction of $L(w;W)R$, and for each formula A we would have $A \in L' \quad \Rightarrow \quad k_0 \Vdash A$ and $A \in R' \quad \Rightarrow \quad k_0 \nVdash A$, then the lemma is a consequence of the *Ktest* rules (all except the K4NW-rule are reversible).

With induction on the length of formula A we will prove that if $k_l \in K$ corresponds to the reduced split sequent $L'(w;W)R'$, then $A \in L'$ implies $k_l \Vdash A$ and $A \in R'$ implies $k_l \nVdash A$.

The cases where A is atomic, a conjunction, a disjunction or an implication are obvious (using fact 6.5.0.5).

Let $A = \Box B$, $A \in L'$ and (as $L'(w;W)R'$ is reduced) $L' = K';M';$. Observe that A will be a member of M' and application of the K4NW-rule will result in a leftmost split sequent containing both A and B. Repeated applications of the K4NW-rule hereafter will result in (leftmost) spliting sequents with the same property.

Let $k_l \rho k_m$ and let $k_m \in K$ correspond to a reduced split sequent $L''(w'', W'')R''$. Now either $w' \in W'$ or $L''(w'', W'')R''$ is the result of (repeated) application of the K4NW-rule. From the observation above infer that in either case $B \in L''$. By induction hypothesis conclude that $k_m \Vdash B$. Which proves that $k_l \Vdash \Box B$.

Note that if no *K4test* rule is applicable for $L'(w;W)R'$ and $w \notin W$, then k_l is an irreflexive terminal node and trivially $k_l \Vdash \Box B$.

If $A = \Box B$ and $A \in R'$ then A will be in T and application of the K4NW-rule (and reduction) will result in a node k_m corresponding to a reduced split sequent $L''(w'';W'')R''$ such that $k_l < k_m$ and $B \in R''$. Using the induction hypothesis we conclude $k_m \nVdash B$ and hence $k_l \nVdash \Box B$. ⊣

6.6.2.7. Theorem. *A split sequent $L(w;W)R$ is closing, using the* K4test *rules, iff* $L \vdash \bigvee R$.

Proof. By combining the previous two lemmas. ⊣

For the differences between *K4test* and *Ktest* one may compare the pseudo-code of *KMtest* with following pseudo-code program, *K4Mtest*, for the algorithm *K4test*.

```
K4Mtest(K, M, N, S, T, U : sequence of formula
                  w : world, W : sequence of world): bool
  if S ≠ ∅
  then let S = ⟨A, S′⟩
        if A ∈ J ∪ K ∪ M ∪ N then true
        else in case A
             atomic : K4Mtest(K, M, N, S′, T, ⟨A, U⟩, w, W)
             ¬B     : K4Mtest(K, M, ⟨B, N⟩, S′, T, U, w, W)
             B ∧ C  : if K4Mtest(K, M, N, ⟨B, S′⟩, T, ⟨A, U⟩, w, W)
                      then K4Mtest(K, M, N, ⟨C, S′⟩, T, ⟨A, U⟩, w, W)
                      else false
             B ∨ C  : K4Mtest(K, M, N, ⟨B, C, S′⟩, T, ⟨A, U⟩, w, W)
             B→C    : KMtest(K, M, ⟨B, N⟩, ⟨C, S′⟩, T, ⟨A, U⟩, w, W)
             □B     : K4Mtest(K, M, N, S′, ⟨A, T⟩, U, w, W)
             ◇B     : K4Mtest(K, M, N, S′, ⟨¬B, T⟩, U, w, W)
  else if N ≠ ∅
        then let N = ⟨A, N′⟩
             if A ∈ T ∪ U then true
             else in case A
                  atomic K4Mtest(⟨A, K⟩, M, N′, , T, U, w, W)
                  ¬B     : K4Mtest(⟨A, K⟩, M, N′, B, T, U, w, W)
                  B ∧ C  : K4Mtest(⟨A, K⟩, M, ⟨A, B, N′⟩, , T, U, w, W)
                  B ∨ C  : if K4Mtest(⟨A, K⟩, M, ⟨B, N⟩, , T, U, w, W)
                           then K4Mtest(⟨A, K⟩, M⟨C, N⟩, , T, U, w, W)
                           else false
                  B→C    : if K4Mtest(⟨A, K⟩, M, N, B, T, U, w, W)
                           then K4Mtest(⟨A, K⟩, M, ⟨C, N⟩, , T, U, w, W)
                           else false
                  □B     : K4Mtest(K, ⟨A, M⟩, N, , T, U, w, W)
                  ◇B     : K4Mtest(K, M, N, , ⟨T, A⟩, U, w, W)
        else if T ≠ ∅
             then let T = ⟨□A, T′⟩ and M* = {B | □B ∈ M}
                                   and w′ = ⟨M ∪ M*, {A}⟩
               if w′ ∈ W ∪ {w′} then K4Mtest(K, M, , , T, U, w, W)
               else K4Mtest(, M, M*, A, , , w′, ⟨w, W⟩)
             else false
```

6.6.3 S4test: an S4 tester

The modal logic **S4** has as its axioms and rules those of **K4** plus the axiom of **T**, $\Box\phi \rightarrow \phi$. A proof that **S4** is complete for finite reflexive and transitive Kripke models can be found in [HC 84]. The tester *S4test* is obtained by replacing the $\Box$L-rule in *K4test* by the Ttest-rule defined above.

6.6.3.8. DEFINITION. *The tester* S4test *has the same rules as* K4test*, except for the $\Box$L-rule that is replaced by the* **T**$\Box$*L-rule.*

6.6.3.9. THEOREM. *A split sequent $L(w; W)R$ is closing, using the* S4test *rules, iff $L \vdash \bigvee R$.*

Proof. The proof is essentially as for theorem 6.6.2.7. The definition of the Kripke model associated with a non-closing sequent has to be changed in such a way that the model is always reflexive. Note that the change in the $\Box L$-rule reflects the addition of the **T** axiom and the reflexivity of the associated models. ⊣

6.6.4 Ltest: an L tester

The modal logic **L** has as its axioms and rules those of **K**, plus the *Löb axiom* $\Box(\Box A \rightarrow A) \rightarrow \Box A$. As in **L** the theorem $\Box A \vdash \Box\Box A$ is derivable, **L** is an extension of **K4**. A proof that **L** is complete for finite, transitive reverse well-founded Kripke models can be found in [Smoryński 85] and [Boolos 93]. The split sequents of the **L** tester *Ltest* will be of the same form as those for **K**.

6.6.4.10. DEFINITION. *The tester* Ltest *has the same rules as* Ktest*, except for the NW-rule that is replaced by the* LNW-rule*:*

$$\frac{K; M;\ \bullet\ ; \Box A, T; U}{; M; M^*, \Box A\ \bullet\ A; ; \quad K; M;\ (w'; W.w)\ ; T; U}$$

where $M^* = \{B \mid \Box B \in M\}$.

6.6.4.11. LEMMA. *The algorithm* Ltest *is deterministic and terminates on the input of any split sequent.*

Proof. The proof is essentially the same as for *K4test* in lemma 6.6.2.3

6.6.4.12. LEMMA. *If a split sequent $L \bullet R$ is closing (by the* Ltest *rules) then $L \vdash \bigvee R$ (in* **L***).*

Proof. For a closed split sequent $L \bullet R$ the lemma is obvious.

As the tableau for a split sequent is a finite tree of split sequents, we can proceed by induction on the depth of the sequent (closed split sequents having depth zero).

All *Ltest* rules can be treated as in the proof of lemma 6.5.0.6, except for the rule LNW, for which we can proceed as in the proof of lemma 6.6.2.4. ⊣

For *Ltest* we define the Kripke model associated with a non-closing split sequent as in definition 6.6.2.5, omitting the looping back rule 3.

6.6.4.13. LEMMA. *If $L \bullet R$ is a non-closing split sequent and K its associated Kripke model, with root k_0, then for each formula A we have $A \in L \quad \Rightarrow \quad k_0 \Vdash A$ and $A \in R \quad \Rightarrow \quad k_0 \nVdash A$.*

Proof. First observe that if $L' \bullet R'$ is the leftmost non-closing reduction of $L \bullet R$, and for each formula A we would have $A \in L' \quad \Rightarrow \quad k_0 \Vdash A$ and $A \in R' \quad \Rightarrow \quad k_0 \nVdash A$, then the lemma is a consequence of the *Ktest* rules (all except the LNW-rule are reversible).

With induction on the length of formula A we will prove that if $k_l \in K$ corresponds to the reduced split sequent $L' \bullet R'$, then $A \in L'$ implies $k_l \Vdash A$ and $A \in R'$ implies $k_l \nVdash A$.

The cases where A is atomic, a conjunction, a disjunction or an implication are obvious (using fact 6.5.0.5).

Let $A = \Box B$, $A \in L'$ and (as $L' \bullet R'$ is reduced) $L' = K'; M';$. Observe that, as in case of the K4NW-rule, A will be a member of M' and application of the LNW-rule will result in a leftmost split sequent containing both A and B. Repeated applications of the LNW-rule hereafter will result in (leftmost) spliting sequents with the same property.

Hence, for the proof of this lemma we can proceed as in the proof of lemma 6.6.2.4 (omitting the case where $w' \in W'$). $\dashv$

The pseudo-code program *LMtest* for the algorithm *Ltest* only differs slightly from the program *K4Mtest* in subsection 6.6.2.

$LMtest(K, M, N, S, T, U$: **sequence of** *formula*): **bool**

  if $S \neq \emptyset$
  then let $S = \langle A, S' \rangle$
      if $A \in J \cup K \cup M \cup N$ **then true**
      else in case A
          atomic **:** $LMtest(K, M, N, S', T, \langle A, U \rangle)$
          $\neg B$ **:** $LMtest(K, M, \langle B, N \rangle, S', T, U)$
          $B \wedge C$ **: if** $LMtest(K, M, N, \langle B, S' \rangle, T, \langle A, U \rangle)$
                then $LMtest(K, M, N, \langle C, S' \rangle, T, \langle A, U \rangle)$
                else false
          $B \vee C$ **:** $LMtest(K, M, N, \langle B, C, S' \rangle, T, \langle A, U \rangle)$
          $B \rightarrow C$ **:** $KMtest(K, M, \langle B, N \rangle, \langle C, S' \rangle, T, \langle A, U \rangle)$
          $\Box B$ **:** $LMtest(K, M, N, S', \langle A, T \rangle, U)$
          $\Diamond B$ **:** $LMtest(K, M, N, S', \langle \neg B, T \rangle, U)$
  else if $N \neq \emptyset$
      then let $N = \langle A, N' \rangle$
          if $A \in T \cup U$ **then true**
          else in case A
              atomic $LMtest(\langle A, K \rangle, M, N', , T, U)$
              $\neg B$ **:** $LMtest(\langle A, K \rangle, M, N', B, T, U)$
              $B \wedge C$ **:** $LMtest(\langle A, K \rangle, M, \langle A, B, N' \rangle, , T, U,)$
              $B \vee C$ **: if** $LMtest(\langle A, K \rangle, M, \langle B, N \rangle, , T, U)$
                    then $LMtest(\langle A, K \rangle, M \langle C, N \rangle, , T, U)$
                    else false
              $B \rightarrow C$ **: if** $LMtest(\langle A, K \rangle, M, N, B, T, U)$
                    then $LMtest(\langle A, K \rangle, M, \langle C, N \rangle, , T, U)$
                    else false
              $\Box B$ **:** $LMtest(K, \langle A, M \rangle, N, , T, U)$
              $\Diamond B$ **:** $LMtest(K, M, N, , \langle T, A \rangle, U)$
      else if $T \neq \emptyset$
          then let $T = \langle \Box A, T' \rangle$ **and** $M^* = \{B \mid \Box B \in M\}$
              if $LMtest(, M, \langle \Box A, M^* \rangle, A, ,)$ **then true**
              else $LMtest(K, M, , , T, U)$
          else false

Appendix A

Computer programs

A.1 Preliminaries

The computer programs in this appendix, written in the programming language C, all make use of a module that contains the types, functions and procedures that are common to the mkDiag program described in section 2.6 and the family of testers treated in Chapter 6. Parts of this module, supporting the understanding of the C-programs in the sequel, are listed below.

In the computer programs in this appendix formulas are represented by (pointers to) structures of the form:

```
struct formType
   { char           type;          /* -,&,|,:,L,M else the atom */
     struct formType *an;
     struct formType *co;
     unsigned       treated : 1;
     unsigned       revisit : 1;
   };

typedef struct formType *formula;
```

The *type* of a formula is denoted by its main connective. The list of possible connectives `-, &, |, :, L, M` corresponds with the list $\neg, \wedge, \vee, \rightarrow, \Box, \Diamond$. If the type character is not in this list, the formula is assumed to be atomic.

If a formula is not atomic, the main subformula(s) is (are) represented in the structure by a pointer to this (these) formula(s). The flags *'treated'* and *'revisit'* are used in the programs to mark the formulas as treated or as to be revisited.

Lists of formulas are represented by simple linked lists:

```
struct flistType
    { formula            element;
      struct flistType    *next;
    };

typedef struct flistType *formlist;
```

The utility module defines procedures for making and printing formulas (either on screen or in a file). For example the procedures *mkatom, mkNegation, mkConjunction* and *mkNecessarily*, to make atomic formulas, negations, conjunctions and necessitations are defined as:

```
formula mkAtom( char c )
{ formula form    = newForm();
  form->type      = findAtom( c );
  form->an        = NULL;
  form->co        = NULL;
  form->treated   = 1;
  return form;
}

formula mkNegation( formula x )
{ formula form = newForm();
  form->type   = '-';
  form->an     = x;
  return form;
}

formula mkConjunction( formula x, formula y )
{ formula form = newForm();
  form->type   = '&';
  form->an     = x;
  form->co     = y;
  return form;
}

formula mkNecessarily( formula x )
{ formula form = newForm();
  form->type   = 'L';
  form->an     = x;
  return form;
}
```

The function `newForm` allocates for a pointer the memory to be used to store the apointed formula structure. The function `findAtom` assigns a numeric character to the type of an atomic formula. This is not really needed for the programs described in this appendix.

A.2 The mkDiag program

A description of the *mkDiag* program can be found in section 2.6. The program makes use of a representation of a *Kripke model* K and an **IpL** *fragment* F to compute the equivalence classes in the fragment in the theory of the model. Hence two formulas ϕ and ψ are equivalent if

$$K \Vdash \phi \leftrightarrow \psi.$$

In the program, the **IpL** fragment F is given by the values of the constants `NEG, DNEG, CON, DIS, IMP` and `MaxMu`, corresponding to the connectives

$\neg, \neg\neg, \wedge, \vee, \rightarrow$ and the maximum level of nesting of the implication in F. The value of a connective-constant will be one or zero, depending on whether the corresponding connective is or is not in F.

The information on the Kripke model K is encoded in the constants `NE, NEM, ALL0, ALL1` and in the functions:

```
eset      comp( eset x);
eset      neg( eset x);
```

The constants `NE, NEM, ALL0` and `ALL1` all are involved in the representation of sets of worlds in K. The constants `NE` and `NEM` are related: `NE` $=$ `NEM` $+ 1$. A subset in K is called an `eset` (element set) in the program and is represented by an array of `NE` integers (`r[0]` to `r[NEM]`), each a binary encoding of a part of the model K. For $0 \leq i \leq$ `NEM` we have `ALL0` as an upper bound, $0 \leq$ `r[i]` $\leq$ `ALL0`. For the last part we have $0 \leq$ `r[NEM]` $\leq$ `ALL1`. In general it will be the case that `ALL1` $\leq$ `ALL0`.

With this information `comp(s)`, the complement of an `eset s`, can simply be calculated.

The *order* in K (the accessibility relation) is encoded in the function `neg`, computing the complement of the predecessor set as defined in definition 2.6.0.2. In section 2.6 it has been explained how the interior of a set in an **IpL** model can be calculated using complements and predecessor sets.

Apart from these procedures and those in the utility module (as described in the previous section), the program makes use of the following procedures:

```
void      classTest( char s, unsigned an, unsigned co, unsigned mu );
unsigned  noSet( eset x );
eset      meet( eset x, eset y );
eset      join( eset x, eset y );
unsigned  Inc( eset x, eset y);
void      fprintSet( FILE *f, eset x );
void      fprintVal( void );
```

The procedure `classTest` makes a new formula, computes the set of worlds in K where this formula is forced and tests (using the function `noSet`) whether or not this set already exists (in the list of formulas and sets `E`). The functions `meet` and `join` compute the meet and join of two sets and `Inc(x, y)` checks whether or not the set `x` is a subset of `y`. The procedure `fprintSet` prints a set (in a readable format) into a (text) file.

The result of the program `mkdiag` is an array `E` of pairs of sets and formulas defined as:

```
struct { eset       set;
         formula    form;
         unsigned   mu;
       } E[Dnr];
```

where `mu` can be used to calculate the nesting of the implication in the formulas and `Dnr` is the maximal number of classes `E` can contain. The number of classes in `E` is denoted by the variable `Emax`.

The output of the program is

1. a text file recording the equivalence classes found and their corresponding subsets in the model;
2. a file with formulas. Depending on one of the run-time parameters for the program, either the formulas in $Diag(F)$ or the representatives of the join-irreducible classes in the diagram are printed in this file;
3. a file describing the order of either the diagram or the set of join-irreducible elements in the diagram (again depending on a run-time parameter).

The last two files are made by the procedure `fprintVal` (not reprinted here).

The procedure *main* below is the main routine in the program *mkDiag*. Its listing is followed by the listings of the most important procedures used in *main* (i.e. *classTest, noSet, meet, join* and *Inc*).

```
main()
{ unsigned i, j, mu;
  char c;

  DiagramStart = 1 - NEG;
  if ( init() )
  { for ( mu=0; mu <= MaxMu; mu++ )
    { printf( "-------------\n mu= %d \n-------------\n", mu );
      fprintf( out, "-------------\n mu= %d \n-------------\n", mu );
      for ( i=DiagramStart; i <= Emax; i++)
      if ( E[i].mu == mu )
      { printf( "%5d  ", i );
        printForm( E[i].form );
        printf( "\n" );
        fprintf(out, "%5d  ", i );
        fprintForm( out, E[i].form );
        fprintSet( out, E[i].set );
      }

      for ( j = 2; j <= Emax; j++ )
      { if ( NEG && E[j].mu == mu - 1 ) classTest('-', j, i, mu);
        if ( DNEG && E[j].mu == mu - 2 ) classTest('d', j, i, mu);
        for ( i = 2; i < j && Emax < Dnr; i++ )
        { if ( E[j].mu == mu )
          { if ( !Inc(E[i].set, E[j].set) )
            { if ( IMP && E[i].mu < mu ) classTest( ':', i, j, mu );
              if ( !Inc(E[j].set, E[i].set) )
              { if ( CON ) classTest( '&', i, j, mu );
                if ( DIS ) classTest( '|', i, j, mu );
              }
            }
          }
          else
          { if ( IMP )
            { if ( E[i].mu == mu-1 && !Inc(E[i].set, E[j].set))
                classTest( ':', i, j, mu );
              if ( !Inc(E[j].set, E[i].set)
                    && ( E[i].mu == mu || E[j].mu == mu-1 ) )
```

```
                classTest( ':', j, i, mu );
            }
          }
        }
      }
    }
    fclose(out);
    fprintVal();
  }
}

void classTest( char s, unsigned an, unsigned co, unsigned mu )
{ unsigned counter, i, n = Emax + 1;
  eset nset;
  void *oldheaptop = getHeapTop();
/* may be the form made is not needed, so remember HeapTop */
  unsigned buz = 1, more, less;
  formula form, fan = E[an].form, fco = E[co].form;

  if ( n == Dnr )
  { printf( "there are too much classes: MaxHeap is too small\n" );
    fprintf( out,
   "there are too much classes: MaxHeap is too small\n" );
    fclose(out);
    exit(4);
  }

  switch ( s )
  { case '-'    :   form = mkNegation( fan );
                    nset = neg( E[an].set ); break;
    case 'd'    :   form = mkNegation(mkNegation( fan ) );
                    nset = neg(neg( E[an].set )); break;
    case '&'    :   form = mkConjunction( fan, fco );
                    nset = meet( E[an].set, E[co].set ); break;
    case '|'    :   form = mkDisjunction( fan, fco );
                    nset = join( E[an].set, E[co].set ); break;
    case ':'    :   form = mkImplication( fan, fco );
                    nset = neg(comp(join( comp(E[an].set),
                    E[co].set))); break;
  }

  if ( noSet(nset) )
  { E[n].form   = form;
    E[n].set    = nset;
    E[n].mu     = mu;
    Emax = n;
    printf( "%5d  ", n );
    printForm( form );
    printf( "\n" );
    fprintf(out, "%5d  ", n );
    fprintForm(out, form );
    fprintSet(out, nset );
    fprintf(out, "\n" );
  }
```

```
  else /* we don't need form anymore */
    setHeapTop( oldheaptop );
}

unsigned   noSet( eset x )
{ unsigned i, j, res=1;

  for (i=0; i<NE && res; i++) res = x.r[i] == 0;
  if ( res ) return E[0].mu > MaxMu;
  res = 1;
  for (i=0; i<NEM && res; i++) res = x.r[i] == ALL0;
  if ( res && x.r[NEM] == ALL1 ) return E[1].mu <= MaxMu ? 0 : 1;
  res = 0;
  for ( i=2; i <= Emax && !res; i++ )
  { res = 1;
    for ( j=0; j < NE && res ; j++) res = (E[i].set.r[j] == x.r[j]);
  }
  return !res;
}

eset  meet( eset x, eset y )
{ eset res;
  unsigned i;

  for (i=0; i < NE; i++) res.r[i] = x.r[i] & y.r[i];
  return res;
}

eset  join( eset x, eset y )
{ eset res;
  unsigned i;

  for (i=0; i < NE; i++) res.r[i] = x.r[i] | y.r[i];
  return res;
}

unsigned Inc( eset x, eset y)
{ unsigned i;

  for (i=0; i<NE && x.r[i] == (x.r[i] & y.r[i]); i++);
  if (i<NE) return 0;
  else return 1;
}
```

A.3 A simple CpL tester

In chapter 6 we calculated the complexity of the algorithm *Ctest*. For comparison we specify an algorithm *Cval* based on truth tables and calculate its complexity. This algorithm assumes a representation of atoms p_i such that calculating i from p_i is simple (i.e. linear in the size of p_i).

For natural numbers i and N, $i \in N$ means that if N is taken as a binary number representing some subset S of $\{0, \ldots, n-1\}$, that $i \in S$.

We will assume that the indices of the atoms in a formula to be tested form some sequence $\{0, \ldots, n-1\}$.

Global $N : I\!\!N$
$Cval(A : \mathit{formula})$: **bool**
  Calculate n the number of atoms in A
  $N = 0$
  while $N < 2^n$ **and** $SubVal(A)$
    $N := N + 1$
  if $N < 2^n$ **then false else true**

$SubVal(A : \mathit{formula})$: **bool**
  in case A
    p_i : **if** $i \in N$ **then true**
      else false
    $\neg B$: **if** $SubVal(B)$ **then false**
      else true
    $B \wedge C$: **if** $SubVal(B)$ **then** $SubVal(C)$
      else false
    $B \vee C$: **if** $SubVal(B)$ **then true**
      else $SubVal(C)$
    $B \rightarrow C$: **if** $SubVal(C)$ **then true**
      else if $SubVal(C)$ **then false**
        else true

To calculate $Cval(\phi)$ for some formula ϕ note that:

1. the main part of $Cval$ needs storage for ϕ and three numbers;
2. the number of atoms in ϕ can be calculated in time and space both linear in $|\phi|$;
3. in $SubVal$ the formula is to be split in its principal subformulas, which takes time in the order of $|\phi|$;
4. $SubVal$ needs space to store three formulas;
5. the number of atoms in ϕ is at most $|\phi|$ and hence there will be at most $2^{|\phi|}$ calls to $SubVal$ (which is also an upperbound of the number of items on stack;

Disregarding the small constants this amounts in an upper bound on the time needed to calculate $Cval(\phi)$ of order $|\phi|.2^{|\phi|}$. Likewise the upper bound on the amount of space needed is of the order $3.|\phi|.2^{|\phi|}$.

A.4 The IpLtest program

The *IpLtest* program is a rather straightforward implementation of the algorithm *IpLtest* (and the pseudo-code program *Itest*) in Chapter 6.

Many of the utilities used in this program do exactly what one would expect them to do.

For example `copy` does make a copy of a formula and `notDisjunct` checks whether or not two formula lists have a common formula. Both `putRight` and `putLeft` add a formula to a formula list, but in the procedure `putLeft`, if the added formula is atomic and does not occurs in the list as an already treated formula, the global flag `LeftChange` is set (compare the rule $pL3$ in the definition of *IpLtest* in Chapter 6. Note that we use a variable `oldvalue` to keep the previous value of `LeftChange` in store if needed.

The procedure `markRevisit` marks the formulas in a list as to be revisited and the function `untreatedFormula` takes an untreated formula out of a formula list (taking value `NULL` if there is no such formula in the list). In the same way `revisitLeftside` has as its result the list of all formulas marked to be revisited, out of a given formula list. And the function `revisitRightside` takes a formula marked to be revisited out of a formula list (again, with value `NULL` if there is no such formula in the list).

The main procedure, `refutable`, has as its input two lists of formulas, `left` and `right` and returns the value zero iff none of the formulas in the list `right` is a consequence (in **IpL**) of the formulas in the list `left`.

If necessary the program writes error messages to an output file (for which a pointer `out` is used).

```
/* FUNCTIONS */
int refutable( formlist left, formlist right )
{formula  form, cform;
 formlist flist;
 int oldval = LeftChange, res;

 if ( notDisjunct( left, right ) ) res = 0;
 else
 { if ( form = untreatedFormula( right ), form )
   { form->treated = 1;
     switch( form->type )
     { case '-' : form->revisit = 1;
                  res = refutable( left, right );
                  form->revisit = 0;
                  break;
       case '&' : res = refutable(left, putRight(form->an,right))
                  ? 1
                  : refutable(left, putRight(form->co,right));
                  break;
       case '|' : res = refutable(left,
                  putRight(form->an,
                  putRight(form->co, right)));
                  break;
       case '=' : res = refutable(left,
                  putRight(
```

```
                 mkConjunction(
                    mkImplication(form->an, form->co),
                    mkImplication(form->co, form->an))
                 right) );
              break;
    case ':' : form->revisit = 1;
              res = refutable( left, right );
              form->revisit = 0;
              break;
  }
  form->treated = 0;
}
else
{ if ( form = untreatedFormula( left ), form )
  { form->treated = 1;
    switch( form->type )
    { case '-' : form->revisit = 1;
                 res = refutable(left,
                 putRight(copy( form->an ),right));
                 form->revisit = 0;
                 break;
      case '&' : res = refutable(putLeft(form->an,
                 putLeft(form->co, left)), right);
                 break;
      case '|' : res = refutable(putLeft(form->an, left), right)
                 ? 1
                 : ( LeftChange = oldval,
                     refutable(putLeft(form->co, left), right)
                   );
                 break;
      case '=' : res = refutable(
                          putLeft(
                             mkConjunction(
                                mkImplication(form->an, form->co),
                                mkImplication(form->co, form->an)),
                             left),
                          right );
                       break;
      case ':' : if ( refutable(putLeft(form->co,left), right) )
                   res = 1;
                 else
                 { LeftChange = oldval;
                   form->revisit = 1;
                   res = refutable(left,
                                      putRight(copy( form->an),
                                   right));
                   form->revisit = 0;
                 }
                 form->revisit = 0;
                 break;
    }
    form->treated = 0;
  }
  else
```

```
     { flist = revisitLeftside( left );
       if ( flist )
       { LeftChange = 0;
         res = refutable( left, right );
         LeftChange = 1;
         markRevisit( flist );
       }
       else
       { form = revisitRightside( right );
         if ( form )
         { form->revisit = 0;
           switch( form->type )
           { case '-' : if ( refutable(putLeft(form->an, left), NULL) )
                        { LeftChange = oldval;
                          res = refutable(left, right );
                        }
                        else res = 0;
                        break;
             case ':' : if ( refutable(putLeft(form->an,left),
                                putRight(form->co,NULL)) )
                        { LeftChange = oldval;
                          res = refutable(left, right);
                        }
                        else res = 0;
                        break;
           }
           form->revisit = 1;
         }
         else res = 1;
       }
     }
   }
 }
 LeftChange = oldval;
 return res;
}
```

A.5 Testers for modal logic

The modal testers described in Chapter 6 have been implemented in one module. Depending on the setting of the constants `T`, `K4`, `S5`, `L` and `Grz` the module is compiled as a tester for **K**, **T**, **K4**, **L**, **S4**, **S5** or Grzegorczyk's logic **K4Grz** (the last two not treated in this thesis).

```
#define T   0  /* Lp->p                                  */
#define K4 0  /* Lp->LLp  S4 == T + K4                  */
#define S5 0  /* S5 == 1  =>  S4 == 1                   */
#define L   0  /* L(Lp->p)->Lp, L == 1   =>  K4 == 1     */
#define Grz 0 /* L(L(p->Lp)->p)->p, Grz == 1 => K4 == 1 */
```

Many of the procedures in the program for the modal testers are the same (at least in principle) as in the *IpLtest* program above. Some noteworthy exceptions are `put`, simply adding a formula to a formula list, and the procedures dealing with (lists of) worlds. Both for worlds and lists of worlds we use pointers:

```
struct worldType
       { formlist      wleft;
         formlist      wright;
       };

typedef struct worldType  *world;

struct wlistType
       { world              element;
         struct wlistType   *next;
       };

typedef struct wlistType  *worldlist;
```

The structure of worlds and their rôle in the algorithm *K4test* has been explained in subsection 6.6.2.

The procedures `newWorld` and `newWorldList` allocate memory needed to store the data of appointed `world` structures and `worldlist` structures. To add worlds to a list of worlds, there is a procedure `addWorld` and to find out whether a given world is in a given list of worlds, there is a procedure `memberworld`.

Again, the main procedure for the program for the formula tester(s) is `refutable(left, right, worlds)`, returning the value 1 iff there is a Kripke model starting with the list of worlds `worlds`, forcing all the formulas in the list `left` and no formula in the list `right`.

```
formlist revisitLeftside( formlist x )
{ formlist res = NULL;
  formula  an, el;

  if ( LeftChange)
  { while ( x )
    { el = x->element;
      if ( el->revisit )
      { an = el->an;
        if (!member(an, res))      /* K */
          res = add( copy(an), res );
#if K4 == 1
        if (!member(el, res))
          res = add( copy(el), res );
#endif
#if K5 == 1
        else res = put(mkPossibly(el), res);
#endif
      }
      x = x->next;
    }
  }
  return res;
}
```

```
formula revisitRightside( formlist x )
{ formula res = NULL;
  while ( !res && x )
  { if ( (x->element)->revisit ) res = x->element;
    else x = x->next;
  }
  return res;
}

formlist addModal( formula f, formlist x)
{ formula g;
  formlist l, y;

  l = x;
  y = add(f->an, NULL);
  while ( l )
  { g = l->element;
    if ( g->revisit && g != f ) y = add(g, y);
    l = l->next;
  }
  return y;
}

int refutable( formlist left, formlist right, worldlist worlds )
{ formula     form,
              cform;
  formlist    flist,
              glist;
  int         oldval = LeftChange,
              putback,
              res = 1;
  world       nwworld;
  worldlist   nwworlds; /* wlist to debug */

  if ( notDisjunct( left, right ) ) res = 0;
  else
  { if ( form = untreatedFormula( right ), form )
    { form->treated = 1;
      switch( form->type )
      { case '-' : res = refutable( put(form->an, left), right, worlds );
                   break;
        case 'L' : form->revisit = 1;
                   res = refutable( left, right, worlds );
                   form->revisit = 0;
                   break;
        case 'M' : res = refutable(
                           put( mkNecessarily(mkNegation(form->an)), left ),
                                right, worlds );
                   break;
        case '&' : res = refutable(left, put(form->an, right), worlds)
                   ? 1
                   : refutable(left, put(form->co, right), worlds);
                   break;
        case '|' : res = refutable(left, put(form->an,
```

```
                                         put(form->co, right)), worlds);
                 break;
      case '=' : res = refutable( left, put(
                                         mkConjunction(
                                            mkImplication(
                                               form->an,
                                               form->co),
                                            mkImplication(
                                               form->co,
                                               form->an)),
                                          right), worlds );
                 break;
      case ':' : res = refutable( put(form->an, left),
                                   put(form->co, right), worlds );
                 break;
     }
     form->treated = 0;
   }
   else
   { if ( form = untreatedFormula( left ), form )
     { form->treated = 1;
       switch( form->type )
       { case '-' : res = refutable(left, put(copy( form->an ), right)
                                         , worlds);
                      break;
#if T == 1
         case 'L' : LeftChange = 1;
                    form->revisit = 1;
                    res = refutable(put(form->an, left), right, worlds);
                    form->revisit = 0;
                    break;
#else
         case 'L' : LeftChange = 1;
                    form->revisit = 1;
                    res = refutable(left, right, worlds);
                    form->revisit = 0;
                    break;
#endif
         case 'M' : res = refutable(
                              left,
                              put(mkNecessarily(mkNegation(form->an)),
                                 right), worlds );
                    break;
         case '&' : res = refutable( put(
                                         form->an,
                                         put(form->co, left)),
                                    right, worlds );
                    break;
         case '|' : res = refutable( put(form->an, left), right, worlds )
                    ? 1
                    : ( LeftChange = oldval,
                    refutable( put(form->co, left), right, worlds ));
                    break;
         case '=' : res = refutable( put(
```

```
                                        mkImplication(form->an, form->co),
                                        put(
                                          mkImplication(form->co, form->an),
                                          left)),
                                      right, worlds);
                          break;
           case ':' : if ( refutable(put(form->co, left), right, worlds ))
                          res = 1;
                          else
                          { LeftChange = oldval;
                            /* form->revisit = 1; */
                            res = refutable(left, put(copy(form->an), right),
                                                          worlds);
                          }
                          break;
           }
           form->treated = 0;
       }
       else
       { if (LeftChange)
         { form = revisitRightside( right );
           if (form)
           { form->revisit = 0;
#if T == 1
             res = member(form->an, right);
             /* if res then the branch will stay open */
             if (!res)
             {
#endif
               flist = revisitLeftside( left );
               LeftChange = 0;
               glist = put(form->an, NULL);
               #if S5 == 1
               glist = addModal(form, right);
#endif
#if L == 1
               if (!member(form, flist)) flist = add(copy(form), flist);
#endif
#if K4 == 0 || L == 1
               res = refutable(flist, glist, worlds);
#else
               nwworld = newWorld();
               nwworld->wleft = flist;
               nwworld->wright = glist;
               res = memberworld(nwworld, worlds);
               /* if res then the branch will stay open */
               if (!res)
               { nwworlds = addWorld(nwworld, worlds);
                 res = refutable(flist, glist, nwworlds);
               }
#endif
#if Grz == 1
               else res = 0;
#endif
```

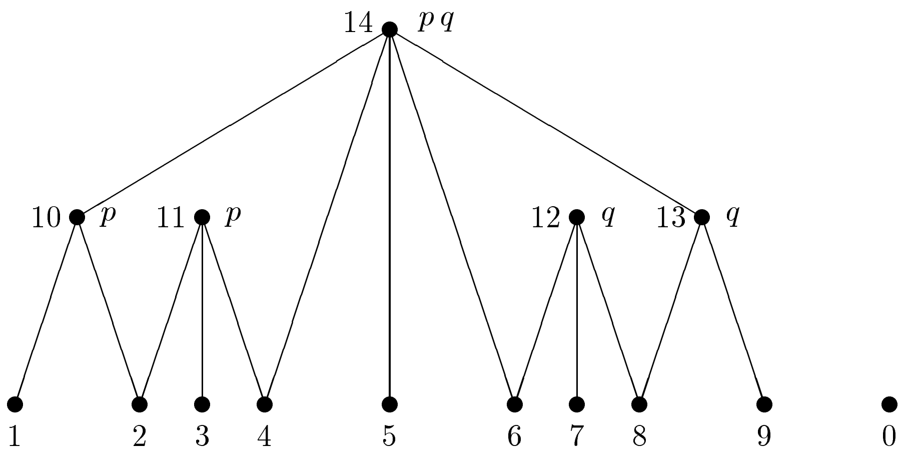
```
#if T == 1
            }
#endif
            if (res)
            { LeftChange = oldval;
              res = refutable(left, right, worlds);
            }
            else res = 0;
            LeftChange = 1;
            form->revisit = 1;
          }
          else res = 1;
        }
        else res = 1;
      }
    }
  }
  LeftChange = oldval;
  return res;
}
```

Appendix B

Output of computer programs

B.1 The diagram of the IpL fragment $[\to, \neg]^2$

The fragment $[\to, \neg]^n$ was treated in subsection 3.5.1. The diagram of $[\to, \neg]^2$, listed below, was computed using the exact Kripke model of $[\wedge, \to, \neg]^2$ (compare figure 20):



31. FIGURE. *The exact model of* $[\wedge, \to, \neg]^2$.

0	$\neg(p\to p)$	$\{\}$
1	$(p\to p)$	$\{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14\}$
2	p	$\{10,11,14\}$
3	q	$\{12,13,14\}$
4	$\neg p$	$\{0,7,12\}$
5	$\neg q$	$\{0,3,11\}$
6	$(p\to q)$	$\{0,5,6,7,8,9,12,13,14\}$
7	$(q\to p)$	$\{0,1,2,3,4,5,10,11,14\}$
8	$(q\to\neg p)$	$\{0,3,7,11,12\}$
9	$\neg\neg p$	$\{1,2,3,4,5,9,10,11,13,14\}$
10	$(\neg p\to q)$	$\{1,2,3,4,5,6,8,9,10,11,12,13,14\}$
11,	$\neg\neg q$	$\{1,5,6,7,8,9,10,12,13,14\}$
12,	$(\neg q\to p)$	$\{1,2,4,5,6,7,8,9,10,11,12,13,14\}$
13,	$(\neg p\to\neg q)$	$\{0,1,2,3,4,5,9,10,11,13,14\}$
14	$(\neg q\to\neg p)$	$\{0,1,5,6,7,8,9,10,12,13,14\}$

15	$\neg(p\to q)$	$\{3, 11\}$
16	$((p\to q)\to p)$	$\{1, 2, 3, 4, 10, 11, 14\}$
17	$((p\to q)\to q)$	$\{1, 2, 3, 4, 10, 11, 12, 13, 14\}$
18	$\neg(q\to p)$	$\{7, 12\}$
19	$((q\to p)\to p)$	$\{6, 7, 8, 9, 10, 11, 12, 13, 14\}$
20	$((q\to p)\to q)$	$\{6, 7, 8, 9, 12, 13, 14\}$
21	$\neg(q\to\neg p)$	$\{1, 5, 9, 10, 13, 14\}$
22	$((q\to\neg p)\to p)$	$\{1, 2, 4, 5, 9, 10, 11, 13, 14\}$
23	$((q\to\neg p)\to q)$	$\{1, 5, 6, 8, 9, 10, 12, 13, 14\}$
24	$(\neg q\to\neg\neg p)$	$\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$
25	$((q\to p)\to\neg(p\to q))$	$\{3, 7, 11, 12\}$
26	$((q\to p)\to((p\to q)\to p))$	$\{1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$
27	$(\neg\neg p\to p)$	$\{0, 6, 7, 10, 11, 12, 14\}$
28	$(\neg\neg p\to q)$	$\{0, 6, 7, 8, 12, 13, 14\}$
29	$(\neg\neg p\to(q\to p))$	$\{0, 1, 2, 3, 4, 5, 6, 7, 10, 11, 12, 14\}$
30	$\neg(\neg p\to q)$	$\{0\}$
31	$((\neg p\to q)\to p)$	$\{0, 10, 11, 14\}$
32	$((\neg p\to q)\to q)$	$\{0, 7, 12, 13, 14\}$
33	$(\neg\neg q\to p)$	$\{0, 2, 3, 4, 10, 11, 14\}$
34	$(\neg\neg q\to q)$	$\{0, 3, 4, 11, 12, 13, 14\}$
35	$(\neg\neg q\to(p\to q))$	$\{0, 3, 4, 5, 6, 7, 8, 9, 11, 12, 13, 14\}$
36	$(\neg\neg q\to(\neg p\to q))$	$\{0, 1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12, 13, 14\}$
37	$((\neg q\to p)\to p)$	$\{0, 3, 10, 11, 14\}$
38	$((\neg q\to p)\to q)$	$\{0, 12, 13, 14\}$
39	$(\neg\neg p\to(\neg q\to p))$	$\{0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$
40	$((\neg p\to\neg q)\to p)$	$\{6, 7, 10, 11, 12, 14\}$
41	$((\neg p\to\neg q)\to q)$	$\{6, 7, 8, 12, 13, 14\}$
42	$((\neg q\to\neg p)\to p)$	$\{2, 3, 4, 10, 11, 14\}$
43	$((\neg q\to\neg p)\to q)$	$\{3, 4, 11, 12, 13, 14\}$
44	$(((p\to q)\to p)\to p)$	$\{0, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$
45	$(\neg\neg p\to((p\to q)\to p))$	$\{0, 1, 2, 3, 4, 6, 7, 10, 11, 12, 14\}$
46	$((\neg p\to q)\to((p\to q)\to p))$	$\{0, 1, 2, 3, 4, 10, 11, 14\}$
47	$((\neg p\to\neg q)\to((p\to q)\to p))$	$\{1, 2, 3, 4, 6, 7, 10, 11, 12, 14\}$
48	$(((p\to q)\to q)\to p)$	$\{0, 5, 10, 11, 14\}$
49	$(\neg\neg p\to((p\to q)\to q))$	$\{0, 1, 2, 3, 4, 6, 7, 8, 10, 11, 12, 13, 14\}$
50	$((\neg p\to q)\to((p\to q)\to q))$	$\{0, 1, 2, 3, 4, 7, 10, 11, 12, 13, 14\}$
51	$(\neg\neg q\to((p\to q)\to q))$	$\{0, 1, 2, 3, 4, 10, 11, 12, 13, 14\}$
52	$((\neg p\to\neg q)\to((p\to q)\to q))$	$\{1, 2, 3, 4, 6, 7, 8, 10, 11, 12, 13, 14\}$
53	$(((q\to p)\to p)\to q)$	$\{0, 5, 12, 13, 14\}$
54	$(\neg\neg p\to((q\to p)\to p))$	$\{0, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$
55	$(\neg\neg q\to((q\to p)\to p))$	$\{0, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$
56	$((\neg q\to p)\to((q\to p)\to p))$	$\{0, 3, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$
57	$((\neg q\to\neg p)\to((q\to p)\to p))$	$\{2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$
58	$(((q\to p)\to p)\to((p\to q)\to q))$	$\{0, 1, 2, 3, 4, 5, 10, 11, 12, 13, 14\}$
59	$(\neg\neg p\to((q\to p)\to q))$	$\{0, 6, 7, 8, 9, 12, 13, 14\}$
60	$(\neg\neg q\to((q\to p)\to q))$	$\{0, 3, 4, 6, 7, 8, 9, 11, 12, 13, 14\}$
61	$((\neg q\to\neg p)\to((q\to p)\to q))$	$\{3, 4, 6, 7, 8, 9, 11, 12, 13, 14\}$
62	$(\neg(q\to\neg p)\to p)$	$\{0, 2, 3, 4, 6, 7, 10, 11, 12, 14\}$
63	$(\neg(q\to\neg p)\to q)$	$\{0, 3, 4, 6, 7, 8, 11, 12, 13, 14\}$
64	$((\neg p\to q)\to\neg(q\to\neg p))$	$\{0, 1, 5, 9, 10, 13, 14\}$
65	$(((q\to\neg p)\to p)\to p)$	$\{0, 3, 6, 7, 10, 11, 12, 14\}$
66	$((\neg p\to q)\to((q\to\neg p)\to p))$	$\{0, 1, 2, 4, 5, 9, 10, 11, 13, 14\}$
67	$(((q\to\neg p)\to q)\to q)$	$\{0, 3, 4, 7, 11, 12, 13, 14\}$
68	$((\neg q\to p)\to((q\to\neg p)\to q))$	$\{0, 1, 5, 6, 8, 9, 10, 12, 13, 14\}$
69	$(\neg\neg p\to((q\to p)\to((p\to q)\to p)))$	$\{0, 1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$
70	$(((p\to q)\to q)\to(\neg\neg p\to p))$	$\{0, 5, 6, 7, 10, 11, 12, 14\}$
71	$(((q\to p)\to p)\to(\neg\neg p\to q))$	$\{0, 5, 6, 7, 8, 12, 13, 14\}$
72	$(((q\to p)\to q)\to(\neg\neg p\to q))$	$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 11, 12, 13, 14\}$
73	$(((q\to p)\to p)\to((\neg p\to q)\to q))$	$\{0, 5, 7, 12, 13, 14\}$
74	$(((q\to p)\to q)\to((\neg p\to q)\to q))$	$\{0, 1, 2, 3, 4, 5, 7, 10, 11, 12, 13, 14\}$
75	$(((p\to q)\to p)\to(\neg\neg q\to p))$	$\{0, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$
76	$(((p\to q)\to q)\to(\neg\neg q\to p))$	$\{0, 2, 3, 4, 5, 10, 11, 14\}$
77	$(((q\to p)\to p)\to(\neg\neg q\to q))$	$\{0, 3, 4, 5, 11, 12, 13, 14\}$
78	$(((p\to q)\to p)\to((\neg q\to p)\to p))$	$\{0, 3, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$
79	$(((p\to q)\to q)\to((\neg q\to p)\to p))$	$\{0, 3, 5, 10, 11, 14\}$
80	$((\neg q\to\neg p)\to((\neg p\to\neg q)\to p))$	$\{2, 3, 4, 6, 7, 10, 11, 12, 14\}$
81	$((\neg q\to\neg p)\to((\neg p\to\neg q)\to q))$	$\{3, 4, 6, 7, 8, 11, 12, 13, 14\}$
82	$(\neg(q\to\neg p)\to(((p\to q)\to q)\to p))$	$\{0, 2, 3, 4, 5, 6, 7, 10, 11, 12, 14\}$

83	$(((q\to\neg p)\to p)\to(((p\to q)\to q)\to p))$	$\{0,3,5,6,7,10,11,12,14\}$
84	$(\neg(q\to\neg p)\to(((q\to p)\to p)\to q))$	$\{0,3,4,5,6,7,8,11,12,13,14\}$
85	$(((q\to\neg p)\to q)\to(((q\to p)\to p)\to q))$	$\{0,3,4,5,7,11,12,13,14\}$
86	$((\neg\neg p\to p)\to q)$	$\{5,8,9,12,13,14\}$
87	$((\neg\neg p\to p)\to((p\to q)\to q))$	$\{1,2,3,4,5,8,9,10,11,12,13,14\}$
88	$((\neg\neg p\to p)\to((q\to p)\to q))$	$\{5,6,7,8,9,12,13,14\}$
89	$((\neg\neg p\to q)\to p)$	$\{1,2,3,4,5,10,11,14\}$
90	$((\neg\neg p\to q)\to q)$	$\{1,2,3,4,5,9,10,11,12,13,14\}$
91	$((\neg\neg p\to(q\to p))\to p)$	$\{9,10,11,13,14\}$
92	$((\neg\neg p\to(q\to p))\to q)$	$\{8,9,12,13,14\}$
93	$((\neg\neg p\to(q\to p))\to((p\to q)\to p))$	$\{1,2,3,4,9,10,11,13,14\}$
94	$((\neg\neg p\to(q\to p))\to((p\to q)\to q))$	$\{1,2,3,4,8,9,10,11,12,13,14\}$
95	$((\neg\neg p\to(q\to p))\to((\neg p\to q)\to p))$	$\{0,9,10,11,13,14\}$
96	$((\neg\neg p\to p)\to((\neg p\to q)\to q))$	$\{0,5,7,8,9,12,13,14\}$
97	$((\neg\neg p\to q)\to((\neg p\to q)\to q))$	$\{0,1,2,3,4,5,7,9,10,11,12,13,14\}$
98	$((\neg\neg p\to(q\to p))\to((\neg p\to q)\to q))$	$\{0,7,8,9,12,13,14\}$
99	$((\neg\neg q\to p)\to p)$	$\{1,5,6,7,8,9,10,11,12,13,14\}$
100	$((\neg\neg q\to p)\to(\neg\neg p\to p))$	$\{0,1,5,6,7,8,9,10,11,12,13,14\}$
101	$((\neg\neg p\to(q\to p))\to(\neg\neg q\to p))$	$\{0,2,3,4,9,10,11,13,14\}$
102	$((\neg\neg q\to q)\to p)$	$\{1,2,5,10,11,14\}$
103	$((\neg\neg q\to q)\to((q\to p)\to p))$	$\{1,2,5,6,7,8,9,10,11,12,13,14\}$
104	$((\neg\neg p\to p)\to(\neg\neg q\to q))$	$\{0,3,4,5,8,9,11,12,13,14\}$
105	$((\neg\neg q\to q)\to(\neg\neg p\to p))$	$\{0,1,2,5,6,7,10,11,12,14\}$
106	$((\neg\neg p\to q)\to(\neg\neg q\to q))$	$\{0,1,2,3,4,5,9,10,11,12,13,14\}$
107	$((\neg\neg p\to(q\to p))\to(\neg\neg q\to q))$	$\{0,3,4,8,9,11,12,13,14\}$
108	$((\neg\neg q\to q)\to((\neg p\to q)\to p))$	$\{0,1,2,5,10,11,14\}$
109	$((\neg\neg q\to(p\to q))\to p)$	$\{1,2,10,11,14\}$
11, 0	$((\neg\neg q\to(p\to q))\to q)$	$\{1,10,12,13,14\}$
11, 1	$((\neg\neg q\to(p\to q))\to((q\to p)\to p))$	$\{1,2,6,7,8,9,10,11,12,13,14\}$
11, 2	$((\neg\neg q\to(p\to q))\to((q\to p)\to q))$	$\{1,6,7,8,9,10,12,13,14\}$
11, 3	$((\neg\neg q\to(p\to q))\to(\neg\neg p\to p))$	$\{0,1,2,6,7,10,11,12,14\}$
11, 4	$((\neg\neg q\to(p\to q))\to(\neg\neg p\to q))$	$\{0,1,6,7,8,10,12,13,14\}$
11, 5	$((\neg\neg q\to(p\to q))\to((\neg p\to q)\to p))$	$\{0,1,2,10,11,14\}$
11, 6	$((\neg\neg q\to(p\to q))\to((\neg p\to q)\to q))$	$\{0,1,7,10,12,13,14\}$
11, 7	$((\neg\neg q\to(\neg p\to q))\to q)$	$\{7,12,13,14\}$
11, 8	$((\neg\neg q\to(\neg p\to q))\to((p\to q)\to q))$	$\{1,2,3,4,7,10,11,12,13,14\}$
11, 9	$((\neg\neg p\to(q\to p))\to((\neg q\to p)\to p))$	$\{0,3,9,10,11,13,14\}$
12, 0	$((\neg\neg q\to p)\to((\neg q\to p)\to p))$	$\{0,1,3,5,6,7,8,9,10,11,12,13,14\}$
12, 1	$((\neg\neg q\to q)\to((\neg q\to p)\to p))$	$\{0,1,2,3,5,10,11,14\}$
12, 2	$((\neg\neg q\to(p\to q))\to((\neg q\to p)\to p))$	$\{0,1,2,3,10,11,14\}$
12, 3	$((\neg\neg p\to p)\to((\neg q\to p)\to q))$	$\{0,5,8,9,12,13,14\}$
12, 4	$((\neg\neg p\to(q\to p))\to((\neg q\to p)\to q))$	$\{0,8,9,12,13,14\}$
12, 5	$((\neg\neg q\to(p\to q))\to((\neg q\to p)\to q))$	$\{0,1,10,12,13,14\}$
12, 6	$((\neg\neg p\to(\neg q\to p))\to p)$	$\{3,10,11,14\}$
12, 7	$((\neg\neg p\to(\neg q\to p))\to((q\to p)\to p))$	$\{3,6,7,8,9,10,11,12,13,14\}$
12, 8	$(((\neg p\to\neg q)\to p)\to((p\to q)\to q))$	$\{0,1,2,3,4,5,8,9,10,11,12,13,14\}$
12, 9	$((\neg\neg p\to q)\to((\neg p\to\neg q)\to p))$	$\{1,2,3,4,5,6,7,10,11,12,14\}$
13, 0	$((\neg\neg q\to q)\to((\neg p\to\neg q)\to p))$	$\{1,2,5,6,7,10,11,12,14\}$
13, 1	$((\neg\neg q\to(p\to q))\to((\neg p\to\neg q)\to p))$	$\{1,2,6,7,10,11,12,14\}$
13, 2	$((\neg\neg p\to(\neg q\to p))\to((\neg p\to\neg q)\to p))$	$\{3,6,7,10,11,12,14\}$
13, 3	$((\neg\neg q\to(p\to q))\to((\neg p\to\neg q)\to q))$	$\{1,6,7,8,10,12,13,14\}$
13, 4	$((\neg\neg p\to(q\to p))\to((\neg q\to\neg p)\to p))$	$\{2,3,4,9,10,11,13,14\}$
13, 5	$(((\neg q\to\neg p)\to q)\to((q\to p)\to p))$	$\{0,1,2,5,6,7,8,9,10,11,12,13,14\}$
13, 6	$((\neg\neg p\to p)\to((\neg q\to\neg p)\to q))$	$\{3,4,5,8,9,11,12,13,14\}$
13, 7	$((\neg\neg p\to(q\to p))\to((\neg q\to\neg p)\to q))$	$\{3,4,8,9,11,12,13,14\}$
13, 8	$(((\neg p\to q)\to p)\to((\neg q\to\neg p)\to q))$	$\{3,4,5,6,7,8,9,11,12,13,14\}$
13, 9	$((\neg\neg q\to(\neg p\to q))\to((\neg q\to\neg p)\to q))$	$\{3,4,7,11,12,13,14\}$
140	$((\neg\neg p\to((p\to q)\to p))\to p)$	$\{5,9,10,11,13,14\}$
141	$((\neg\neg p\to((p\to q)\to p))\to((q\to p)\to p))$	$\{5,6,7,8,9,10,11,12,13,14\}$
142	$((\neg\neg p\to((p\to q)\to p))\to((\neg p\to q)\to p))$	$\{0,5,9,10,11,13,14\}$
143	$((\neg\neg p\to((p\to q)\to p))\to(\neg\neg q\to p))$	$\{0,2,3,4,5,9,10,11,13,14\}$
144	$((\neg\neg p\to((p\to q)\to p))\to((\neg q\to p)\to p))$	$\{0,3,5,9,10,11,13,14\}$
145	$((\neg\neg p\to((p\to q)\to p))\to((\neg q\to\neg p)\to p))$	$\{2,3,4,5,9,10,11,13,14\}$
146	$((\neg\neg p\to(q\to p))\to((\neg p\to q)\to((p\to q)\to p)))$	$\{0,1,2,3,4,9,10,11,13,14\}$
147	$(((\neg p\to q)\to((p\to q)\to p))\to((\neg q\to\neg p)\to p))$	$\{2,3,4,5,6,7,8,9,10,11,12,13,14\}$
148	$((\neg\neg p\to((p\to q)\to q))\to p)$	$\{5,10,11,14\}$
149	$((\neg\neg p\to((p\to q)\to q))\to q)$	$\{5,9,12,13,14\}$
150	$((\neg\neg p\to((p\to q)\to q))\to((\neg p\to q)\to q))$	$\{0,5,7,9,12,13,14\}$

151	$((\neg\neg p\to((p\to q)\to q))\to(\neg\neg q\to q))$	$\{0,3,4,5,9,11,12,13,14\}$
152	$((\neg\neg p\to((p\to q)\to q))\to((\neg q\to p)\to q))$	$\{0,5,9,12,13,14\}$
153	$((\neg\neg p\to((p\to q)\to q))\to((\neg p\to\neg q)\to p))$	$\{5,6,7,10,11,12,14\}$
154	$((\neg\neg p\to((p\to q)\to q))\to((\neg q\to\neg p)\to p))$	$\{2,3,4,5,10,11,14\}$
155	$((\neg\neg p\to((p\to q)\to q))\to((\neg q\to\neg p)\to q))$	$\{3,4,5,9,11,12,13,14\}$
156	$(((\neg p\to q)\to((p\to q)\to q))\to q)$	$\{5,6,8,9,12,13,14\}$
157	$((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))$	$\{0,1,2,3,4,5,7,8,9,10,11,12,13,14\}$
158	$((\neg\neg p\to(q\to p))\to((\neg p\to q)\to((p\to q)\to q)))$	$\{0,1,2,3,4,7,8,9,10,11,12,13,14\}$
159	$(((\neg p\to q)\to((p\to q)\to q))\to(\neg\neg q\to q))$	$\{0,3,4,5,6,8,9,11,12,13,14\}$
160	$(((\neg p\to q)\to((p\to q)\to q))\to((\neg q\to p)\to q))$	$\{0,5,6,8,9,12,13,14\}$
161	$(((\neg p\to q)\to((p\to q)\to q))\to((\neg q\to\neg p)\to q))$	$\{3,4,5,6,8,9,11,12,13,14\}$
162	$((\neg\neg p\to(q\to p))\to(\neg\neg q\to((p\to q)\to q)))$	$\{0,1,2,3,4,8,9,10,11,12,13,14\}$
163	$((\neg\neg q\to(p\to q))\to(((q\to p)\to p)\to q))$	$\{0,1,5,10,12,13,14\}$
164	$((\neg\neg p\to((q\to p)\to p))\to q)$	$\{5,12,13,14\}$
165	$((\neg\neg p\to((q\to p)\to p))\to((p\to q)\to q))$	$\{1,2,3,4,5,10,11,12,13,14\}$
166	$((\neg\neg q\to(p\to q))\to(\neg\neg p\to((q\to p)\to p)))$	$\{0,1,2,6,7,8,9,10,11,12,13,14\}$
167	$((\neg\neg p\to((q\to p)\to p))\to((\neg p\to\neg q)\to q))$	$\{5,6,7,8,12,13,14\}$
168	$((\neg\neg p\to((q\to p)\to p))\to((\neg q\to\neg p)\to q))$	$\{3,4,5,11,12,13,14\}$
169	$((\neg\neg p\to((q\to p)\to p))\to((\neg p\to\neg q)\to((p\to q)\to q)))$	$\{1,2,3,4,5,6,7,8,10,11,12,13,14\}$
170	$((\neg\neg q\to((q\to p)\to p))\to p)$	$\{1,5,10,11,14\}$
171	$((\neg\neg q\to((q\to p)\to p))\to(\neg\neg p\to p))$	$\{0,1,5,6,7,10,11,12,14\}$
172	$((\neg\neg q\to((q\to p)\to p))\to((\neg p\to q)\to p))$	$\{0,1,5,10,11,14\}$
173	$((\neg\neg q\to((q\to p)\to p))\to((\neg q\to p)\to p))$	$\{0,1,3,5,10,11,14\}$
174	$((\neg\neg q\to((q\to p)\to p))\to((\neg p\to\neg q)\to p))$	$\{1,5,6,7,10,11,12,14\}$
175	$(((\neg q\to p)\to((q\to p)\to p))\to p)$	$\{1,2,4,5,10,11,14\}$
176	$(((\neg q\to p)\to((q\to p)\to p))\to(\neg\neg p\to p))$	$\{0,1,2,4,5,6,7,10,11,12,14\}$
177	$(((\neg q\to p)\to((q\to p)\to p))\to((\neg p\to q)\to p))$	$\{0,1,2,4,5,10,11,14\}$
178	$((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))$	$\{0,1,2,3,5,6,7,8,9,10,11,12,13,14\}$
179	$((\neg\neg q\to(p\to q))\to((\neg q\to p)\to((q\to p)\to p)))$	$\{0,1,2,3,6,7,8,9,10,11,12,13,14\}$
180	$(((\neg q\to p)\to((q\to p)\to p))\to((\neg p\to\neg q)\to p))$	$\{1,2,4,5,6,7,10,11,12,14\}$
181	$((\neg\neg q\to(p\to q))\to(\neg\neg p\to((q\to p)\to q)))$	$\{0,1,6,7,8,9,10,12,13,14\}$
182	$((\neg\neg q\to((q\to p)\to q))\to q)$	$\{1,5,10,12,13,14\}$
183	$((\neg\neg q\to((q\to p)\to q))\to(\neg\neg p\to q))$	$\{0,1,5,6,7,8,10,12,13,14\}$
184	$((\neg\neg q\to((q\to p)\to q))\to((\neg p\to q)\to q))$	$\{0,1,5,7,10,12,13,14\}$
185	$((\neg\neg q\to((q\to p)\to q))\to((\neg p\to\neg q)\to q))$	$\{1,5,6,7,8,10,12,13,14\}$
186	$((\neg(q\to\neg p)\to p)\to p)$	$\{1,5,9,10,11,13,14\}$
187	$((\neg(q\to\neg p)\to p)\to((\neg p\to q)\to p))$	$\{0,1,5,9,10,11,13,14\}$
188	$((\neg(q\to\neg p)\to p)\to((\neg q\to p)\to p))$	$\{0,1,3,5,9,10,11,13,14\}$
189	$((\neg(q\to\neg p)\to q)\to q)$	$\{1,5,9,10,12,13,14\}$
190	$((\neg(q\to\neg p)\to q)\to((\neg p\to q)\to q))$	$\{0,1,5,7,9,10,12,13,14\}$
191	$((\neg(q\to\neg p)\to q)\to((\neg q\to p)\to q))$	$\{0,1,5,9,10,12,13,14\}$
192	$((\neg\neg q\to q)\to(((q\to\neg p)\to p)\to p))$	$\{0,1,2,3,5,6,7,10,11,12,14\}$
193	$((\neg\neg q\to(p\to q))\to(((q\to\neg p)\to p)\to p))$	$\{0,1,2,3,6,7,10,11,12,14\}$
194	$((\neg\neg q\to((q\to p)\to p))\to(((q\to\neg p)\to p)\to p))$	$\{0,1,3,5,6,7,10,11,12,14\}$
195	$((\neg\neg p\to p)\to(((q\to\neg p)\to q)\to q))$	$\{0,3,4,5,7,8,9,11,12,13,14\}$
196	$((\neg\neg p\to(q\to p))\to(((q\to\neg p)\to q)\to q))$	$\{0,3,4,7,8,9,11,12,13,14\}$
197	$((\neg\neg p\to((p\to q)\to q))\to(((q\to\neg p)\to q)\to q))$	$\{0,3,4,5,7,9,11,12,13,14\}$
198	$((((q\to p)\to p)\to(\neg\neg p\to q))\to q)$	$\{1,2,3,4,9,10,11,12,13,14\}$
199	$((((q\to p)\to p)\to(\neg\neg p\to q))\to((\neg p\to q)\to q))$	$\{0,1,2,3,4,7,9,10,11,12,13,14\}$
200	$((((q\to p)\to p)\to(\neg\neg p\to q))\to(\neg\neg q\to q))$	$\{0,1,2,3,4,9,10,11,12,13,14\}$
201	$((((q\to p)\to q)\to(\neg\neg p\to q))\to q)$	$\{9,12,13,14\}$
202	$((((q\to p)\to q)\to(\neg\neg p\to q))\to((\neg p\to q)\to q))$	$\{0,7,9,12,13,14\}$
203	$((((q\to p)\to q)\to(\neg\neg p\to q))\to(\neg\neg q\to q))$	$\{0,3,4,9,11,12,13,14\}$
204	$((((q\to p)\to q)\to(\neg\neg p\to q))\to((\neg q\to p)\to q))$	$\{0,9,12,13,14\}$
205	$((((q\to p)\to q)\to(\neg\neg p\to q))\to((\neg q\to\neg p)\to q))$	$\{3,4,9,11,12,13,14\}$
206	$((((q\to p)\to q)\to(\neg\neg p\to q))\to(((q\to\neg p)\to q)\to q))$	$\{0,3,4,7,9,11,12,13,14\}$
207	$((((q\to p)\to p)\to((\neg p\to q)\to q))\to q)$	$\{1,2,3,4,6,8,9,10,11,12,13,14\}$
208	$((((q\to p)\to p)\to((\neg p\to q)\to q))\to(\neg\neg q\to q))$	$\{0,1,2,3,4,6,8,9,10,11,12,13,14\}$
209	$((((q\to p)\to q)\to((\neg p\to q)\to q))\to q)$	$\{6,8,9,12,13,14\}$
210	$((((q\to p)\to q)\to((\neg p\to q)\to q))\to(\neg\neg q\to q))$	$\{0,3,4,6,8,9,11,12,13,14\}$
211,	$((((q\to p)\to q)\to((\neg p\to q)\to q))\to((\neg q\to p)\to q))$	$\{0,6,8,9,12,13,14\}$
212,	$((((q\to p)\to q)\to((\neg p\to q)\to q))\to((\neg q\to\neg p)\to q))$	$\{3,4,6,8,9,11,12,13,14\}$
213,	$((((p\to q)\to p)\to(\neg\neg q\to p))\to p)$	$\{1,10,11,14\}$
214	$((((p\to q)\to p)\to(\neg\neg q\to p))\to((q\to p)\to p))$	$\{1,6,7,8,9,10,11,12,13,14\}$
215	$((((p\to q)\to p)\to(\neg\neg q\to p))\to(\neg\neg p\to p))$	$\{0,1,6,7,10,11,12,14\}$
216	$((((p\to q)\to p)\to(\neg\neg q\to p))\to((\neg p\to q)\to p))$	$\{0,1,10,11,14\}$
217	$((((p\to q)\to p)\to(\neg\neg q\to p))\to((\neg q\to p)\to p))$	$\{0,1,3,10,11,14\}$
218	$((((p\to q)\to p)\to(\neg\neg q\to p))\to((\neg p\to\neg q)\to p))$	$\{1,6,7,10,11,12,14\}$

219	$((((p\to q)\to p)\to(\neg\neg q\to p))\to(\neg\neg p\to((q\to p)\to p)))$	$\{0,1,6,7,8,9,10,11,12,13,14\}$
220	$((((p\to q)\to p)\to(\neg\neg q\to p))\to((\neg q\to p)\to((q\to p)\to p)))$	$\{0,1,3,6,7,8,9,10,11,12,13,14\}$
221	$((((p\to q)\to p)\to(\neg\neg q\to p))\to(((q\to\neg p)\to p)\to p))$	$\{0,1,3,6,7,10,11,12,14\}$
222	$((((p\to q)\to p)\to((\neg q\to p)\to p))\to p)$	$\{1,2,4,10,11,14\}$
223	$((((p\to q)\to p)\to((\neg q\to p)\to p))\to((q\to p)\to p))$	$\{1,2,4,6,7,8,9,10,11,12,13,14\}$
224	$((((p\to q)\to p)\to((\neg q\to p)\to p))\to(\neg\neg p\to p))$	$\{0,1,2,4,6,7,10,11,12,14\}$
225	$((((p\to q)\to p)\to((\neg q\to p)\to p))\to((\neg p\to q)\to p))$	$\{0,1,2,4,10,11,14\}$
226	$((((p\to q)\to p)\to((\neg q\to p)\to p))\to((\neg p\to\neg q)\to p))$	$\{1,2,4,6,7,10,11,12,14\}$
227	$((((p\to q)\to p)\to((\neg q\to p)\to p))\to(\neg\neg p\to((q\to p)\to p)))$	$\{0,1,2,4,6,7,8,9,10,11,12,13,14\}$
228	$((\neg\neg p\to((p\to q)\to q))\to((\neg q\to\neg p)\to((\neg p\to\neg q)\to p)))$	$\{2,3,4,5,6,7,10,11,12,14\}$
229	$((\neg\neg p\to((q\to p)\to p))\to((\neg q\to\neg p)\to((\neg p\to\neg q)\to q)))$	$\{3,4,5,6,7,8,11,12,13,14\}$
230	$((\neg(q\to\neg p)\to(((p\to q)\to q)\to p))\to p)$	$\{1,9,10,11,13,14\}$
231	$((\neg(q\to\neg p)\to(((p\to q)\to q)\to p))\to((\neg p\to q)\to p))$	$\{0,1,9,10,11,13,14\}$
232	$((\neg(q\to\neg p)\to(((p\to q)\to q)\to p))\to((\neg q\to p)\to p))$	$\{0,1,3,9,10,11,13,14\}$
233	$((((q\to\neg p)\to p)\to(((p\to q)\to q)\to p))\to p)$	$\{1,2,4,9,10,11,13,14\}$
234	$((((q\to\neg p)\to p)\to(((p\to q)\to q)\to p))\to((\neg p\to q)\to p))$	$\{0,1,2,4,9,10,11,13,14\}$
235	$((\neg(q\to\neg p)\to(((q\to p)\to p)\to q))\to q)$	$\{1,9,10,12,13,14\}$
236	$((\neg(q\to\neg p)\to(((q\to p)\to p)\to q))\to((\neg p\to q)\to q))$	$\{0,1,7,9,10,12,13,14\}$
237	$((\neg(q\to\neg p)\to(((q\to p)\to p)\to q))\to((\neg q\to p)\to q))$	$\{0,1,9,10,12,13,14\}$
238	$((((q\to\neg p)\to q)\to(((q\to p)\to p)\to q))\to q)$	$\{1,6,8,9,10,12,13,14\}$
239	$((((q\to\neg p)\to q)\to(((q\to p)\to p)\to q))\to((\neg q\to p)\to q))$	$\{0,1,6,8,9,10,12,13,14\}$
240	$((\neg\neg q\to(p\to q))\to((\neg\neg p\to p)\to q))$	$\{1,5,8,9,10,12,13,14\}$
241	$((\neg\neg q\to(\neg p\to q))\to((\neg\neg p\to p)\to q))$	$\{5,7,8,9,12,13,14\}$
242	$(((\neg q\to\neg p)\to((q\to p)\to q))\to((\neg\neg p\to p)\to q))$	$\{0,1,5,8,9,10,12,13,14\}$
243	$(((\neg q\to p)\to((q\to\neg p)\to q))\to((\neg\neg p\to p)\to q))$	$\{3,4,5,7,8,9,11,12,13,14\}$
244	$((\neg\neg q\to(\neg p\to q))\to((\neg\neg p\to p)\to((p\to q)\to q)))$	$\{1,2,3,4,5,7,8,9,10,11,12,13,14\}$
245	$((\neg\neg q\to(\neg p\to q))\to((\neg\neg p\to q)\to q))$	$\{1,2,3,4,5,7,9,10,11,12,13,14\}$
246	$((\neg\neg q\to q)\to((\neg\neg p\to(q\to p))\to p))$	$\{1,2,5,9,10,11,13,14\}$
247	$((\neg\neg q\to(p\to q))\to((\neg\neg p\to(q\to p))\to p))$	$\{1,2,9,10,11,13,14\}$
248	$((\neg\neg p\to(\neg q\to p))\to((\neg\neg p\to(q\to p))\to p))$	$\{3,9,10,11,13,14\}$
249	$(((\neg q\to\neg p)\to q)\to((\neg\neg p\to(q\to p))\to p))$	$\{0,1,2,5,9,10,11,13,14\}$
250	$((\neg\neg q\to(p\to q))\to((\neg\neg p\to(q\to p))\to q))$	$\{1,8,9,10,12,13,14\}$
251	$((\neg\neg q\to(\neg p\to q))\to((\neg\neg p\to(q\to p))\to q))$	$\{7,8,9,12,13,14\}$
252	$(((\neg q\to p)\to((q\to\neg p)\to q))\to((\neg\neg p\to(q\to p))\to q))$	$\{3,4,7,8,9,11,12,13,14\}$
253	$((\neg\neg q\to(\neg p\to q))\to((\neg\neg p\to(q\to p))\to((p\to q)\to q)))$	$\{1,2,3,4,7,8,9,10,11,12,13,14\}$
254	$((\neg\neg q\to(p\to q))\to((\neg\neg p\to(q\to p))\to((\neg p\to q)\to p)))$	$\{0,1,2,9,10,11,13,14\}$
255	$((\neg\neg q\to(p\to q))\to((\neg\neg p\to p)\to((\neg p\to q)\to q)))$	$\{0,1,5,7,8,9,10,12,13,14\}$
256	$((\neg\neg q\to(p\to q))\to((\neg\neg p\to(q\to p))\to((\neg p\to q)\to q)))$	$\{0,1,7,8,9,10,12,13,14\}$
257	$((\neg\neg p\to(\neg q\to p))\to((\neg\neg q\to p)\to p))$	$\{1,3,5,6,7,8,9,10,11,12,13,14\}$
258	$((\neg\neg p\to(\neg q\to p))\to((\neg\neg q\to q)\to p))$	$\{1,2,3,5,10,11,14\}$
259	$(((\neg p\to q)\to((q\to\neg p)\to p))\to((\neg\neg q\to q)\to p))$	$\{1,2,3,5,6,7,10,11,12,14\}$
260	$((\neg\neg p\to(\neg q\to p))\to((\neg\neg q\to q)\to((q\to p)\to p)))$	$\{1,2,3,5,6,7,8,9,10,11,12,13,14\}$
261	$((\neg\neg p\to(\neg q\to p))\to((\neg\neg q\to(p\to q))\to p))$	$\{1,2,3,10,11,14\}$
262	$(((\neg p\to q)\to((q\to\neg p)\to p))\to((\neg\neg q\to(p\to q))\to p))$	$\{1,2,3,6,7,10,11,12,14\}$
263	$((\neg\neg q\to(\neg p\to q))\to((\neg\neg q\to(p\to q))\to q))$	$\{1,7,10,12,13,14\}$
264	$((\neg\neg p\to(\neg q\to p))\to((\neg\neg q\to(p\to q))\to((q\to p)\to p)))$	$\{1,2,3,6,7,8,9,10,11,12,13,14\}$
265	$((\neg\neg p\to((p\to q)\to q))\to((\neg\neg q\to(\neg p\to q))\to q))$	$\{5,7,9,12,13,14\}$
266	$((\neg\neg p\to((q\to p)\to p))\to((\neg\neg q\to(\neg p\to q))\to q))$	$\{5,7,12,13,14\}$
267	$((\neg\neg p\to((q\to p)\to q))\to((\neg\neg q\to(\neg p\to q))\to q))$	$\{1,2,3,4,5,7,10,11,12,13,14\}$
268	$((\neg\neg q\to((q\to p)\to q))\to((\neg\neg q\to(\neg p\to q))\to q))$	$\{1,5,7,10,12,13,14\}$
269	$((\neg(q\to\neg p)\to q)\to((\neg\neg q\to(\neg p\to q))\to q))$	$\{1,5,7,9,10,12,13,14\}$
270	$((((q\to p)\to p)\to(\neg\neg p\to q))\to((\neg\neg q\to(\neg p\to q))\to q))$	$\{1,2,3,4,7,9,10,11,12,13,14\}$
271	$((((q\to p)\to q)\to(\neg\neg p\to q))\to((\neg\neg q\to(\neg p\to q))\to q))$	$\{7,9,12,13,14\}$
272	$((\neg(q\to\neg p)\to(((q\to p)\to p)\to q))\to((\neg\neg q\to(\neg p\to q))\to q))$	$\{1,7,9,10,12,13,14\}$
273	$((\neg\neg q\to q)\to((\neg\neg p\to(q\to p))\to((\neg q\to p)\to p)))$	$\{0,1,2,3,5,9,10,11,13,14\}$
274	$((\neg\neg q\to(p\to q))\to((\neg\neg p\to(q\to p))\to((\neg q\to p)\to p)))$	$\{0,1,2,3,9,10,11,13,14\}$
275	$((\neg\neg q\to(p\to q))\to((\neg\neg p\to(q\to p))\to((\neg q\to p)\to q)))$	$\{0,1,8,9,10,12,13,14\}$
276	$((\neg\neg p\to((p\to q)\to p))\to((\neg\neg p\to(\neg q\to p))\to p))$	$\{3,5,9,10,11,13,14\}$
277	$(((\neg p\to q)\to((p\to q)\to p))\to((\neg\neg p\to(\neg q\to p))\to p))$	$\{3,5,6,7,8,9,10,11,12,13,14\}$
278	$((\neg\neg p\to((p\to q)\to q))\to((\neg\neg p\to(\neg q\to p))\to p))$	$\{3,5,10,11,14\}$
279	$((\neg\neg q\to((q\to p)\to p))\to((\neg\neg p\to(\neg q\to p))\to p))$	$\{1,3,5,10,11,14\}$
280	$((\neg(q\to\neg p)\to p)\to((\neg\neg p\to(\neg q\to p))\to p))$	$\{1,3,5,9,10,11,13,14\}$
281	$((((p\to q)\to p)\to(\neg\neg q\to p))\to((\neg\neg p\to(\neg q\to p))\to p))$	$\{1,3,10,11,14\}$
282	$((((p\to q)\to q)\to(\neg\neg q\to p))\to((\neg\neg p\to(\neg q\to p))\to p))$	$\{1,3,6,7,8,9,10,11,12,13,14\}$
283	$((\neg(q\to\neg p)\to(((p\to q)\to q)\to p))\to((\neg\neg p\to(\neg q\to p))\to p))$	$\{1,3,9,10,11,13,14\}$
284	$((\neg\neg p\to((p\to q)\to q))\to((\neg\neg p\to(\neg q\to p))\to((\neg p\to\neg q)\to p)))$	$\{3,5,6,7,10,11,12,14\}$
285	$((\neg\neg q\to((q\to p)\to p))\to((\neg\neg p\to(\neg q\to p))\to((\neg p\to\neg q)\to p)))$	$\{1,3,5,6,7,10,11,12,14\}$
286	$((((p\to q)\to p)\to(\neg\neg q\to p))\to((\neg\neg p\to(\neg q\to p))\to$	

	$((\neg p\to\neg q)\to p)))$	$\{1,3,6,7,10,11,12,14\}$
287	$((\neg\neg p\to((p\to q)\to q))\to((\neg\neg q\to(\neg p\to q))\to((\neg q\to\neg p)\to q)))$	$\{3,4,5,7,9,11,12,13,14\}$
288	$((\neg\neg p\to((q\to p)\to p))\to((\neg\neg q\to(\neg p\to q))\to((\neg q\to\neg p)\to q)))$	$\{3,4,5,7,11,12,13,14\}$
289	$((((q\to p)\to q)\to(\neg\neg p\to q))\to((\neg\neg q\to(\neg p\to q))\to$ $((\neg q\to\neg p)\to q)))$	$\{3,4,7,9,11,12,13,14\}$
290	$((\neg\neg q\to(\neg p\to q))\to((\neg\neg q\to(p\to q))\to((\neg\neg p\to p)\to q)))$	$\{1,5,7,8,9,10,12,13,14\}$
291	$((\neg\neg p\to(\neg q\to p))\to((\neg\neg q\to q)\to((\neg\neg p\to(q\to p))\to p)))$	$\{1,2,3,5,9,10,11,13,14\}$
292	$((\neg\neg p\to(\neg q\to p))\to((\neg\neg q\to(p\to q))\to((\neg\neg p\to(q\to p))\to p)))$	$\{1,2,3,9,10,11,13,14\}$
293	$((\neg\neg q\to(\neg p\to q))\to((\neg\neg q\to(p\to q))\to((\neg\neg p\to(q\to p))\to q)))$	$\{1,7,8,9,10,12,13,14\}$
294	$(((\neg\neg p\to p)\to q)\to q)$	$\{0,1,2,3,4,6,7,10,11,12,13,14\}$
295	$(((\neg\neg p\to p)\to q)\to(((q\to p)\to p)\to q))$	$\{0,1,2,3,4,5,6,7,10,11,12,13,14\}$
296	$(((\neg\neg p\to p)\to((p\to q)\to q))\to q)$	$\{0,6,7,12,13,14\}$
297	$(((\neg\neg p\to p)\to((p\to q)\to q))\to(\neg\neg q\to q))$	$\{0,3,4,6,7,11,12,13,14\}$
298	$(((\neg\neg p\to p)\to((p\to q)\to q))\to(((q\to p)\to p)\to q))$	$\{0,5,6,7,12,13,14\}$
299	$(((\neg\neg p\to p)\to((p\to q)\to q))\to(((q\to p)\to p)\to(\neg\neg q\to q)))$	$\{0,3,4,5,6,7,11,12,13,14\}$
300	$(((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))$	$\{0,1,2,3,4,5,6,7,9,10,11,12,13,14\}$
301	$(((\neg\neg p\to p)\to((\neg p\to q)\to q))\to q)$	$\{1,2,3,4,6,10,11,12,13,14\}$
302	$(((\neg\neg p\to p)\to((\neg p\to q)\to q))\to(\neg\neg q\to q))$	$\{0,1,2,3,4,6,10,11,12,13,14\}$
303	$(((\neg\neg p\to p)\to((\neg p\to q)\to q))\to(((q\to p)\to p)\to q))$	$\{0,1,2,3,4,5,6,10,11,12,13,14\}$
304	$(((\neg\neg p\to p)\to((\neg p\to q)\to q))\to((\neg\neg p\to q)\to q))$	$\{1,2,3,4,5,6,9,10,11,12,13,14\}$
305	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to q)$	$\{6,8,12,13,14\}$
306	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to((p\to q)\to q))$	$\{1,2,3,4,6,8,10,11,12,13,14\}$
307	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to(\neg\neg q\to q))$	$\{0,3,4,6,8,11,12,13,14\}$
308	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to((\neg q\to p)\to q))$	$\{0,6,8,12,13,14\}$
309	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to((\neg q\to\neg p)\to q))$	$\{3,4,6,8,11,12,13,14\}$
310	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to(\neg\neg q\to((p\to q)\to q)))$	$\{0,1,2,3,4,6,8,10,11,12,13,14\}$
311,	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to(((q\to p)\to p)\to q))$	$\{0,5,6,8,12,13,14\}$
312,	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to(((q\to p)\to p)\to((p\to q)\to q)))$	$\{0,1,2,3,4,5,6,8,10,11,12,13,14\}$
313,	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to(((q\to p)\to p)\to(\neg\neg q\to q)))$	$\{0,3,4,5,6,8,11,12,13,14\}$
314	$(((\neg\neg p\to(q\to p))\to((\neg p\to q)\to q))\to q)$	$\{1,2,3,4,5,6,10,11,12,13,14\}$
315	$(((\neg\neg q\to q)\to p)\to p)$	$\{0,3,4,6,7,8,9,10,11,12,13,14\}$
316	$(((\neg\neg q\to q)\to p)\to(((p\to q)\to p)\to p))$	$\{0,3,4,5,6,7,8,9,10,11,12,13,14\}$
317	$(((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))$	$\{0,1,3,4,5,6,7,8,9,10,11,12,13,14\}$
318	$(((\neg\neg q\to q)\to((q\to p)\to p))\to p)$	$\{0,3,4,10,11,14\}$
319	$(((\neg\neg q\to q)\to((q\to p)\to p))\to(\neg\neg p\to p))$	$\{0,3,4,6,7,10,11,12,14\}$
320	$(((\neg\neg q\to q)\to((q\to p)\to p))\to(((p\to q)\to q)\to p))$	$\{0,3,4,5,10,11,14\}$
321	$(((\neg\neg q\to q)\to((q\to p)\to p))\to(((p\to q)\to q)\to(\neg\neg p\to p)))$	$\{0,3,4,5,6,7,10,11,12,14\}$
322	$(((\neg\neg q\to q)\to((q\to p)\to p))\to((\neg\neg p\to(q\to p))\to p))$	$\{0,3,4,9,10,11,13,14\}$
323	$(((\neg\neg p\to p)\to(\neg\neg q\to q))\to q)$	$\{1,6,7,10,12,13,14\}$
324	$(((\neg\neg p\to p)\to(\neg\neg q\to q))\to((p\to q)\to q))$	$\{1,2,3,4,6,7,10,11,12,13,14\}$
325	$(((\neg\neg p\to p)\to(\neg\neg q\to q))\to((\neg p\to q)\to q))$	$\{0,1,6,7,10,12,13,14\}$
326	$(((\neg\neg p\to p)\to(\neg\neg q\to q))\to(((q\to p)\to p)\to q))$	$\{0,1,5,6,7,10,12,13,14\}$
327	$(((\neg\neg p\to p)\to(\neg\neg q\to q))\to((\neg\neg p\to q)\to q))$	$\{1,2,3,4,5,6,7,9,10,11,12,13,14\}$
328	$(((\neg\neg q\to q)\to(\neg\neg p\to p))\to p)$	$\{3,4,9,10,11,13,14\}$
329	$(((\neg\neg q\to q)\to(\neg\neg p\to p))\to((q\to p)\to p))$	$\{3,4,6,7,8,9,10,11,12,13,14\}$
330	$(((\neg\neg q\to q)\to(\neg\neg p\to p))\to(((p\to q)\to q)\to p))$	$\{0,3,4,5,9,10,11,13,14\}$
331	$(((\neg\neg q\to q)\to(\neg\neg p\to p))\to((\neg\neg q\to p)\to p))$	$\{1,3,4,5,6,7,8,9,10,11,12,13,14\}$
332	$(((\neg\neg p\to p)\to((\neg p\to q)\to q))\to((\neg\neg p\to q)\to(\neg\neg q\to q)))$	$\{0,1,2,3,4,5,6,9,10,11,12,13,14\}$
333	$(((\neg\neg p\to(q\to p))\to(\neg\neg q\to q))\to q)$	$\{1,5,6,7,10,12,13,14\}$
334	$(((\neg\neg p\to(q\to p))\to(\neg\neg q\to q))\to((p\to q)\to q))$	$\{1,2,3,4,5,6,7,10,11,12,13,14\}$
335	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to((\neg\neg q\to(p\to q))\to q))$	$\{1,6,8,10,12,13,14\}$
336	$(((\neg\neg q\to(p\to q))\to(\neg\neg p\to p))\to p)$	$\{3,4,5,9,10,11,13,14\}$
337	$(((\neg\neg q\to(p\to q))\to(\neg\neg p\to p))\to((q\to p)\to p))$	$\{3,4,5,6,7,8,9,10,11,12,13,14\}$
338	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to p)$	$\{2,4,10,11,14\}$
339	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to((q\to p)\to p))$	$\{2,4,6,7,8,9,10,11,12,13,14\}$
340	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to(\neg\neg p\to p))$	$\{0,2,4,6,7,10,11,12,14\}$
341	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to((\neg p\to q)\to p))$	$\{0,2,4,10,11,14\}$
342	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to((\neg p\to\neg q)\to p))$	$\{2,4,6,7,10,11,12,14\}$
343	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to(((p\to q)\to p)\to p))$	$\{0,2,4,5,6,7,8,9,10,11,12,13,14\}$
344	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to(((p\to q)\to q)\to p))$	$\{0,2,4,5,10,11,14\}$
345	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to(\neg\neg p\to((q\to p)\to p)))$	$\{0,2,4,6,7,8,9,10,11,12,13,14\}$
346	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to(((p\to q)\to q)\to(\neg\neg p\to p)))$	$\{0,2,4,5,6,7,10,11,12,14\}$
347	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to((\neg\neg p\to(q\to p))\to p))$	$\{2,4,9,10,11,13,14\}$
348	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to((\neg\neg p\to(q\to p))\to((\neg p\to q)\to p)))$	$\{0,2,4,9,10,11,13,14\}$
349	$(((\neg\neg q\to q)\to((\neg q\to p)\to p))\to p)$	$\{4,6,7,8,9,10,11,12,13,14\}$
350	$(((\neg\neg q\to q)\to((\neg q\to p)\to p))\to(\neg\neg p\to p))$	$\{0,4,6,7,8,9,10,11,12,13,14\}$
351	$(((\neg\neg q\to q)\to((\neg q\to p)\to p))\to(((p\to q)\to p)\to p))$	$\{0,4,5,6,7,8,9,10,11,12,13,14\}$
352	$(((\neg\neg q\to q)\to((\neg q\to p)\to p))\to((\neg\neg q\to p)\to p))$	$\{1,4,5,6,7,8,9,10,11,12,13,14\}$

353	$(((\neg\neg q\to q)\to((\neg q\to p)\to p))\to((\neg\neg q\to p)\to(\neg\neg p\to p)))$	$\{0,1,4,5,6,7,8,9,10,11,12,13,14\}$
354	$(((\neg\neg q\to(p\to q))\to((\neg q\to p)\to p))\to p)$	$\{4,5,6,7,8,9,10,11,12,13,14\}$
355	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to((\neg\neg q\to(p\to q))\to((\neg q\to p)\to q)))$	$\{0,1,6,8,10,12,13,14\}$
356	$((((\neg p\to\neg q)\to p)\to((p\to q)\to q))\to q)$	$\{6,7,12,13,14\}$
357	$((((\neg p\to\neg q)\to p)\to((p\to q)\to q))\to((\neg q\to\neg p)\to q))$	$\{3,4,6,7,11,12,13,14\}$
358	$((((\neg q\to\neg p)\to q)\to((q\to p)\to p))\to p)$	$\{3,4,10,11,14\}$
359	$((((\neg q\to\neg p)\to q)\to((q\to p)\to p))\to((\neg p\to\neg q)\to p))$	$\{3,4,6,7,10,11,12,14\}$
360	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to((\neg\neg p\to((p\to q)\to p))\to p))$	$\{2,4,5,9,10,11,13,14\}$
361	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to((\neg\neg p\to((p\to q)\to p))\to ((q\to p)\to p)))$	$\{2,4,5,6,7,8,9,10,11,12,13,14\}$
362	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to((\neg\neg p\to((p\to q)\to p))\to ((\neg p\to q)\to p)))$	$\{0,2,4,5,9,10,11,13,14\}$
363	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to((\neg\neg p\to((p\to q)\to q))\to p))$	$\{2,4,5,10,11,14\}$
364	$((((\neg q\to\neg p)\to q)\to((q\to p)\to p))\to((\neg\neg p\to((p\to q)\to q))\to p))$	$\{3,4,5,10,11,14\}$
365	$(((\neg\neg p\to p)\to((p\to q)\to q))\to((\neg\neg p\to((p\to q)\to q))\to q))$	$\{0,5,6,7,9,12,13,14\}$
366	$(((\neg\neg p\to p)\to(\neg\neg q\to q))\to((\neg\neg p\to((p\to q)\to q))\to q))$	$\{1,5,6,7,9,10,12,13,14\}$
367	$((((\neg p\to\neg q)\to p)\to((p\to q)\to q))\to((\neg\neg p\to((p\to q)\to q))\to q))$	$\{5,6,7,9,12,13,14\}$
368	$(((\neg\neg p\to p)\to((\neg q\to\neg p)\to q))\to((\neg\neg p\to((p\to q)\to q))\to q))$	$\{0,1,5,6,7,9,10,12,13,14\}$
369	$(((\neg\neg p\to p)\to((p\to q)\to q))\to((\neg\neg p\to((p\to q)\to q))\to(\neg\neg q\to q)))$	$\{0,3,4,5,6,7,9,11,12,13,14\}$
370	$(((\neg\neg q\to p)\to((\neg q\to p)\to p))\to((\neg\neg p\to((p\to q)\to q))\to ((\neg p\to\neg q)\to p)))$	$\{2,4,5,6,7,10,11,12,14\}$
371	$((((\neg q\to\neg p)\to q)\to((q\to p)\to p))\to((\neg\neg p\to((p\to q)\to q))\to ((\neg p\to\neg q)\to p)))$	$\{3,4,5,6,7,10,11,12,14\}$
372	$((((\neg p\to\neg q)\to p)\to((p\to q)\to q))\to((\neg\neg p\to((p\to q)\to q))\to ((\neg q\to\neg p)\to q)))$	$\{3,4,5,6,7,9,11,12,13,14\}$
373	$(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to q)$	$\{6,12,13,14\}$
374	$(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to(\neg\neg q\to q))$	$\{0,3,4,6,11,12,13,14\}$
375	$(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg q\to p)\to q))$	$\{0,6,12,13,14\}$
376	$(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg q\to\neg p)\to q))$	$\{3,4,6,11,12,13,14\}$
377	$(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to(((q\to p)\to p)\to q))$	$\{0,5,6,12,13,14\}$
378	$(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to(((q\to p)\to p)\to (\neg\neg q\to q)))$	$\{0,3,4,5,6,11,12,13,14\}$
379	$(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg\neg q\to(p\to q))\to q))$	$\{1,6,10,12,13,14\}$
380	$(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg\neg q\to(p\to q))\to ((\neg q\to p)\to q)))$	$\{0,1,6,10,12,13,14\}$
381	$(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg\neg p\to ((p\to q)\to q))\to q))$	$\{5,6,9,12,13,14\}$
382	$(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg\neg p\to((p\to q)\to q))\to (\neg\neg q\to q)))$	$\{0,3,4,5,6,9,11,12,13,14\}$
383	$(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg\neg p\to((p\to q)\to q))\to ((\neg q\to p)\to q)))$	$\{0,5,6,9,12,13,14\}$
384	$(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg\neg p\to((p\to q)\to q))\to ((\neg q\to\neg p)\to q)))$	$\{3,4,5,6,9,11,12,13,14\}$
385	$(((\neg\neg p\to(q\to p))\to((\neg p\to q)\to((p\to q)\to q)))\to q)$	$\{5,6,12,13,14\}$
386	$(((\neg\neg p\to(q\to p))\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg q\to\neg p)\to q))$	$\{3,4,5,6,11,12,13,14\}$
387	$(((\neg\neg p\to(q\to p))\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg\neg q\to (p\to q))\to q))$	$\{1,5,6,10,12,13,14\}$
388	$(((\neg\neg p\to(q\to p))\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg\neg q\to (\neg p\to q))\to q))$	$\{5,6,7,12,13,14\}$
389	$(((\neg\neg p\to(q\to p))\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg\neg q\to(p\to q))\to ((\neg q\to p)\to q)))$	$\{0,1,5,6,10,12,13,14\}$
390	$(((\neg\neg p\to(q\to p))\to((\neg p\to q)\to((p\to q)\to q)))\to((\neg\neg q\to(\neg p\to q))\to ((\neg q\to\neg p)\to q)))$	$\{3,4,5,6,7,11,12,13,14\}$
391	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to((\neg\neg q\to(p\to q))\to (((q\to p)\to p)\to q)))$	$\{0,1,5,6,8,10,12,13,14\}$
392	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to((\neg\neg p\to((q\to p)\to p))\to q))$	$\{5,6,8,12,13,14\}$
393	$(((\neg\neg p\to((p\to q)\to q))\to((\neg p\to q)\to q))\to((\neg\neg p\to ((q\to p)\to p))\to q))$	$\{1,2,3,4,5,6,8,10,11,12,13,14\}$
394	$(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to((\neg\neg p\to((q\to p)\to p))\to ((\neg q\to\neg p)\to q)))$	$\{3,4,5,6,8,11,12,13,14\}$
395	$(((\neg\neg q\to q)\to((q\to p)\to p))\to((\neg\neg q\to((q\to p)\to p))\to p))$	$\{0,1,3,4,5,10,11,14\}$
396	$(((\neg\neg q\to q)\to(\neg\neg p\to p))\to((\neg\neg q\to((q\to p)\to p))\to p))$	$\{1,3,4,5,9,10,11,13,14\}$
397	$(((\neg\neg q\to q)\to((\neg p\to\neg q)\to p))\to((\neg\neg q\to((q\to p)\to p))\to p))$	$\{0,1,3,4,5,9,10,11,13,14\}$
398	$((((\neg q\to\neg p)\to q)\to((q\to p)\to p))\to((\neg\neg q\to((q\to p)\to p))\to p))$	$\{1,3,4,5,10,11,14\}$
399	$(((\neg\neg q\to q)\to((q\to p)\to p))\to((\neg\neg q\to((q\to p)\to p))\to(\neg\neg p\to p)))$	$\{0,1,3,4,5,6,7,10,11,12,14\}$
400	$((((\neg q\to\neg p)\to q)\to((q\to p)\to p))\to((\neg\neg q\to((q\to p)\to p))\to ((\neg p\to\neg q)\to p)))$	$\{1,3,4,5,6,7,10,11,12,14\}$
401	$(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to p)$	$\{4,10,11,14\}$

402 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to(\neg\neg p\to p))$ $\{0,4,6,7,10,11,12,14\}$

403 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to((\neg p\to q)\to p))$ $\{0,4,10,11,14\}$

404 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to((\neg p\to\neg q)\to p))$ $\{4,6,7,10,11,12,14\}$

405 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to(((p\to q)\to q)\to p))$ $\{0,4,5,10,11,14\}$

406 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to(((p\to q)\to q)\to (\neg\neg p\to p)))$ $\{0,4,5,6,7,10,11,12,14\}$

407 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to((\neg\neg p\to(q\to p))\to p))$ $\{4,9,10,11,13,14\}$

408 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to((\neg\neg p\to(q\to p))\to ((\neg p\to q)\to p)))$ $\{0,4,9,10,11,13,14\}$

409 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to((\neg\neg p\to ((p\to q)\to p))\to p))$ $\{4,5,9,10,11,13,14\}$

410 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to ((\neg\neg p\to((p\to q)\to p))\to((\neg p\to q)\to p)))$ $\{0,4,5,9,10,11,13,14\}$

411, $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to ((\neg\neg p\to((p\to q)\to q))\to p))$ $\{4,5,10,11,14\}$

412, $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to ((\neg\neg p\to((p\to q)\to q))\to((\neg p\to\neg q)\to p)))$ $\{4,5,6,7,10,11,12,14\}$

413, $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to ((\neg\neg q\to((q\to p)\to p))\to p))$ $\{1,4,5,10,11,14\}$

414 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to ((\neg\neg q\to((q\to p)\to p))\to(\neg\neg p\to p)))$ $\{0,1,4,5,6,7,10,11,12,14\}$

415 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to ((\neg\neg q\to((q\to p)\to p))\to((\neg p\to q)\to p)))$ $\{0,1,4,5,10,11,14\}$

416 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to ((\neg\neg q\to((q\to p)\to p))\to((\neg p\to\neg q)\to p)))$ $\{1,4,5,6,7,10,11,12,14\}$

417 $(((\neg\neg p\to q)\to((\neg p\to q)\to q))\to((\neg\neg q\to ((q\to p)\to q))\to q))$ $\{1,5,6,8,10,12,13,14\}$

418 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to ((\neg(q\to\neg p)\to p)\to p))$ $\{1,4,5,9,10,11,13,14\}$

419 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to ((\neg(q\to\neg p)\to p)\to((\neg p\to q)\to p)))$ $\{0,1,4,5,9,10,11,13,14\}$

420 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to ((\neg(q\to\neg p)\to q)\to q))$ $\{1,5,6,9,10,12,13,14\}$

421 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to ((\neg(q\to\neg p)\to q)\to((\neg q\to p)\to q)))$ $\{0,1,5,6,9,10,12,13,14\}$

422 $(((\neg\neg p\to p)\to q)\to((((q\to p)\to p)\to(\neg\neg p\to q))\to q))$ $\{0,1,2,3,4,6,7,9,10,11,12,13,14\}$

423 $(((\neg\neg p\to p)\to((\neg p\to q)\to q))\to((((q\to p)\to p)\to(\neg\neg p\to q))\to q))$ $\{1,2,3,4,6,9,10,11,12,13,14\}$

424 $(((\neg\neg p\to p)\to(\neg\neg q\to q))\to((((q\to p)\to p)\to(\neg\neg p\to q))\to q))$ $\{1,2,3,4,6,7,9,10,11,12,13,14\}$

425 $(((\neg\neg p\to p)\to((\neg p\to q)\to q))\to((((q\to p)\to p)\to (\neg\neg p\to q))\to(\neg\neg q\to q)))$ $\{0,1,2,3,4,6,9,10,11,12,13,14\}$

426 $(((\neg\neg p\to p)\to((p\to q)\to q))\to((((q\to p)\to q)\to(\neg\neg p\to q))\to q))$ $\{0,6,7,9,12,13,14\}$

427 $(((\neg\neg p\to p)\to(\neg\neg q\to q))\to((((q\to p)\to q)\to(\neg\neg p\to q))\to q))$ $\{1,6,7,9,10,12,13,14\}$

428 $((((\neg p\to\neg q)\to p)\to((p\to q)\to q))\to((((q\to p)\to q)\to (\neg\neg p\to q))\to q))$ $\{6,7,9,12,13,14\}$

429 $(((\neg\neg p\to p)\to((\neg q\to\neg p)\to q))\to((((q\to p)\to q)\to(\neg\neg p\to q))\to q))$ $\{0,1,6,7,9,10,12,13,14\}$

430 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to ((((q\to p)\to q)\to(\neg\neg p\to q))\to q))$ $\{6,9,12,13,14\}$

431 $(((\neg\neg p\to p)\to(((q\to\neg p)\to q)\to q))\to((((q\to p)\to q)\to (\neg\neg p\to q))\to q))$ $\{1,6,9,10,12,13,14\}$

432 $(((\neg\neg p\to p)\to((p\to q)\to q))\to((((q\to p)\to q)\to (\neg\neg p\to q))\to(\neg\neg q\to q)))$ $\{0,3,4,6,7,9,11,12,13,14\}$

433 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to ((((q\to p)\to q)\to(\neg\neg p\to q))\to(\neg\neg q\to q)))$ $\{0,3,4,6,9,11,12,13,14\}$

434 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to ((((q\to p)\to q)\to(\neg\neg p\to q))\to((\neg q\to p)\to q)))$ $\{0,6,9,12,13,14\}$

435 $(((\neg\neg p\to p)\to(((q\to\neg p)\to q)\to q))\to ((((q\to p)\to q)\to(\neg\neg p\to q))\to((\neg q\to p)\to q)))$ $\{0,1,6,9,10,12,13,14\}$

436 $((((\neg p\to\neg q)\to p)\to((p\to q)\to q))\to ((((q\to p)\to q)\to(\neg\neg p\to q))\to((\neg q\to\neg p)\to q)))$ $\{3,4,6,7,9,11,12,13,14\}$

437 $(((\neg\neg p\to p)\to((\neg p\to q)\to((p\to q)\to q)))\to ((((q\to p)\to q)\to(\neg\neg p\to q))\to((\neg q\to\neg p)\to q)))$ $\{3,4,6,9,11,12,13,14\}$

438 $(((\neg\neg q\to q)\to p)\to((((p\to q)\to p)\to(\neg\neg q\to p))\to p))$ $\{0,1,3,4,6,7,8,9,10,11,12,13,14\}$

439 $(((\neg\neg q\to q)\to((q\to p)\to p))\to((((p\to q)\to p)\to(\neg\neg q\to p))\to p))$ $\{0,1,3,4,10,11,14\}$

440 $(((\neg\neg q\to q)\to(\neg\neg p\to p))\to((((p\to q)\to p)\to(\neg\neg q\to p))\to p))$ $\{1,3,4,9,10,11,13,14\}$

441 $(((\neg\neg q\to q)\to((\neg p\to q)\to p))\to((((p\to q)\to p)\to(\neg\neg q\to p))\to p))$ $\{1,3,4,6,7,8,9,10,11,12,13,14\}$

442 $(((\neg\neg q\to q)\to((\neg q\to p)\to p))\to((((p\to q)\to p)\to(\neg\neg q\to p))\to p))$ $\{1,4,6,7,8,9,10,11,12,13,14\}$

443 $(((\neg\neg q\to q)\to((\neg p\to\neg q)\to p))\to((((p\to q)\to p)\to(\neg\neg q\to p))\to p))$ $\{0,1,3,4,9,10,11,13,14\}$

444 $((((\neg q\to\neg p)\to q)\to((q\to p)\to p))\to((((p\to q)\to p)\to$

$(\neg\neg q\to p))\to p))$ $\{1,3,4,10,11,14\}$

445 $(((\neg\neg q\to q)\to((\neg q\to p)\to$
$((q\to p)\to p)))\to((((p\to q)\to p)\to(\neg\neg q\to p))\to p))$ $\{1,4,10,11,14\}$

446 $(((\neg\neg q\to q)\to(((q\to\neg p)\to p)\to p))\to$
$((((p\to q)\to p)\to(\neg\neg q\to p))\to p))$ $\{1,4,9,10,11,13,14\}$

447 $(((\neg\neg q\to q)\to((q\to p)\to p))\to$
$((((p\to q)\to p)\to(\neg\neg q\to p))\to(\neg\neg p\to p)))$ $\{0,1,3,4,6,7,10,11,12,14\}$

448 $(((\neg\neg q\to q)\to((\neg q\to p)\to p))\to$
$((((p\to q)\to p)\to(\neg\neg q\to p))\to(\neg\neg p\to p)))$ $\{0,1,4,6,7,8,9,10,11,12,13,14\}$

449 $(((\neg\neg q\to q)\to((\neg q\to p)\to$
$((q\to p)\to p)))\to((((p\to q)\to p)\to(\neg\neg q\to p))\to(\neg\neg p\to p)))$ $\{0,1,4,6,7,10,11,12,14\}$

450 $(((\neg\neg q\to q)\to((\neg q\to p)\to$
$((q\to p)\to p)))\to((((p\to q)\to p)\to(\neg\neg q\to p))\to((\neg p\to q)\to p)))$ $\{0,1,4,10,11,14\}$

451 $(((\neg\neg q\to q)\to(((q\to\neg p)\to p)\to p))\to$
$((((p\to q)\to p)\to(\neg\neg q\to p))\to((\neg p\to q)\to p)))$ $\{0,1,4,9,10,11,13,14\}$

452 $((((\neg q\to\neg p)\to q)\to((q\to p)\to p))\to$
$((((p\to q)\to p)\to(\neg\neg q\to p))\to((\neg p\to\neg q)\to p)))$ $\{1,3,4,6,7,10,11,12,14\}$

453 $(((\neg\neg q\to q)\to((\neg q\to p)\to((q\to p)\to p)))\to$
$((((p\to q)\to p)\to(\neg\neg q\to p))\to((\neg p\to\neg q)\to p)))$ $\{1,4,6,7,10,11,12,14\}$

454 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to q)$ $\{8,12,13,14\}$

455 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to((p\to q)\to q))$ $\{1,2,3,4,8,10,11,12,13,14\}$

456 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to((\neg p\to q)\to q))$ $\{0,7,8,12,13,14\}$

457 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to(\neg\neg q\to q))$ $\{0,3,4,8,11,12,13,14\}$

458 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to((\neg q\to p)\to q))$ $\{0,8,12,13,14\}$

459 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to((\neg q\to\neg p)\to q))$ $\{3,4,8,11,12,13,14\}$

460 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to((\neg p\to q)\to((p\to q)\to q)))$ $\{0,1,2,3,4,7,8,10,11,12,13,14\}$

461 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to(\neg\neg q\to((p\to q)\to q)))$ $\{0,1,2,3,4,8,10,11,12,13,14\}$

462 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to(((q\to p)\to p)\to q))$ $\{0,5,8,12,13,14\}$

463 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$(((q\to p)\to p)\to((p\to q)\to q)))$ $\{0,1,2,3,4,5,8,10,11,12,13,14\}$

464 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to(((q\to\neg p)\to q)\to q))$ $\{0,3,4,7,8,11,12,13,14\}$

465 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$(((q\to p)\to p)\to((\neg p\to q)\to q)))$ $\{0,5,7,8,12,13,14\}$

466 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$(((q\to p)\to q)\to((\neg p\to q)\to q)))$ $\{0,1,2,3,4,5,7,8,10,11,12,13,14\}$

467 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$(((q\to p)\to p)\to(\neg\neg q\to q)))$ $\{0,3,4,5,8,11,12,13,14\}$

468 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$(((q\to\neg p)\to q)\to(((q\to p)\to p)\to q)))$ $\{0,3,4,5,7,8,11,12,13,14\}$

469 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to((\neg\neg q\to(p\to q))\to q))$ $\{1,8,10,12,13,14\}$

470 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg q\to(p\to q))\to((\neg p\to q)\to q)))$ $\{0,1,7,8,10,12,13,14\}$

471 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to((\neg\neg q\to(\neg p\to q))\to q))$ $\{7,8,12,13,14\}$

472 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg q\to(\neg p\to q))\to((p\to q)\to q)))$ $\{1,2,3,4,7,8,10,11,12,13,14\}$

473 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg q\to(p\to q))\to((\neg q\to p)\to q)))$ $\{0,1,8,10,12,13,14\}$

474 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg q\to(\neg p\to q))\to((\neg q\to\neg p)\to q)))$ $\{3,4,7,8,11,12,13,14\}$

475 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg q\to(p\to q))\to(((q\to p)\to p)\to q)))$ $\{0,1,5,8,10,12,13,14\}$

476 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg p\to((q\to p)\to p))\to q))$ $\{5,8,12,13,14\}$

477 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg p\to((q\to p)\to p))\to((p\to q)\to q)))$ $\{1,2,3,4,5,8,10,11,12,13,14\}$

478 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg p\to((q\to p)\to p))\to((\neg q\to\neg p)\to q)))$ $\{3,4,5,8,11,12,13,14\}$

479 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg q\to((q\to p)\to q))\to q))$ $\{1,5,8,10,12,13,14\}$

480 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg q\to((q\to p)\to q))\to((\neg p\to q)\to q)))$ $\{0,1,5,7,8,10,12,13,14\}$

481 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg q\to(\neg p\to q))\to((\neg\neg q\to(p\to q))\to q)))$ $\{1,7,8,10,12,13,14\}$

482 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg p\to((q\to p)\to p))\to((\neg\neg q\to(\neg p\to q))\to q)))$ $\{5,7,8,12,13,14\}$

483 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg p\to((q\to p)\to q))\to((\neg\neg q\to(\neg p\to q))\to q)))$ $\{1,2,3,4,5,7,8,10,11,12,13,14\}$

484 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$

$((\neg\neg q\to((q\to p)\to q))\to((\neg\neg q\to(\neg p\to q))\to q)))$ $\{1,5,7,8,10,12,13,14\}$

485 $((((\neg\neg p\to p)\to q)\to((\neg\neg p\to q)\to q))\to$
$((\neg\neg p\to((q\to p)\to p))\to((\neg\neg q\to(\neg p\to q))\to((\neg q\to\neg p)\to q))))$ $\{3,4,5,7,8,11,12,13,14\}$

486 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to p)$ $\{2,10,11,14\}$

487 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to((q\to p)\to p))$ $\{2,6,7,8,9,10,11,12,13,14\}$

488 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to(\neg\neg p\to p))$ $\{0,2,6,7,10,11,12,14\}$

489 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to((\neg p\to q)\to p))$ $\{0,2,10,11,14\}$

490 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to((\neg q\to p)\to p))$ $\{0,2,3,10,11,14\}$

491 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to((\neg p\to\neg q)\to p))$ $\{2,6,7,10,11,12,14\}$

492 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to(((p\to q)\to p)\to p))$ $\{0,2,5,6,7,8,9,10,11,12,13,14\}$

493 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to(((p\to q)\to q)\to p))$ $\{0,2,5,10,11,14\}$

494 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to(\neg\neg p\to((q\to p)\to p)))$ $\{0,2,6,7,8,9,10,11,12,13,14\}$

495 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to((\neg q\to p)\to((q\to p)\to p)))$ $\{0,2,3,6,7,8,9,10,11,12,13,14\}$

496 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to(((q\to\neg p)\to p)\to p))$ $\{0,2,3,6,7,10,11,12,14\}$

497 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to(((p\to q)\to q)\to(\neg\neg p\to p)))$ $\{0,2,5,6,7,10,11,12,14\}$

498 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$
$(((p\to q)\to p)\to((\neg q\to p)\to p)))$ $\{0,2,3,5,6,7,8,9,10,11,12,13,14\}$

499 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$
$(((p\to q)\to q)\to((\neg q\to p)\to p)))$ $\{0,2,3,5,10,11,14\}$

500 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$
$(((q\to\neg p)\to p)\to(((p\to q)\to q)\to p)))$ $\{0,2,3,5,6,7,10,11,12,14\}$

501 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to((\neg\neg p\to(q\to p))\to p))$ $\{2,9,10,11,13,14\}$

502 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$
$((\neg\neg p\to(q\to p))\to((\neg p\to q)\to p)))$ $\{0,2,9,10,11,13,14\}$

503 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$
$((\neg\neg p\to(q\to p))\to((\neg q\to p)\to p)))$ $\{0,2,3,9,10,11,13,14\}$

504 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to((\neg\neg p\to(\neg q\to p))\to p))$ $\{2,3,10,11,14\}$

505 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$
$((\neg\neg p\to(\neg q\to p))\to((q\to p)\to p)))$ $\{2,3,6,7,8,9,10,11,12,13,14\}$

506 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$
$((\neg\neg p\to(\neg q\to p))\to((\neg p\to\neg q)\to p)))$ $\{2,3,6,7,10,11,12,14\}$

507 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$
$((\neg\neg p\to((p\to q)\to p))\to p))$ $\{2,5,9,10,11,13,14\}$

508 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$
$((\neg\neg p\to((p\to q)\to p))\to((q\to p)\to p)))$ $\{2,5,6,7,8,9,10,11,12,13,14\}$

509 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$
$((\neg\neg p\to((p\to q)\to p))\to((\neg p\to q)\to p)))$ $\{0,2,5,9,10,11,13,14\}$

510 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$
$((\neg\neg p\to((p\to q)\to p))\to((\neg q\to p)\to p)))$ $\{0,2,3,5,9,10,11,13,14\}$

511, $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$
$((\neg\neg p\to((p\to q)\to q))\to p))$ $\{2,5,10,11,14\}$

512, $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$
$((\neg\neg p\to((p\to q)\to q))\to((\neg p\to\neg q)\to p)))$ $\{2,5,6,7,10,11,12,14\}$

513, $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$
$((\neg\neg p\to(\neg q\to p))\to((\neg\neg p\to(q\to p))\to p)))$ $\{2,3,9,10,11,13,14\}$

514 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$
$((\neg\neg p\to((p\to q)\to p))\to((\neg\neg p\to(\neg q\to p))\to p)))$ $\{2,3,5,9,10,11,13,14\}$

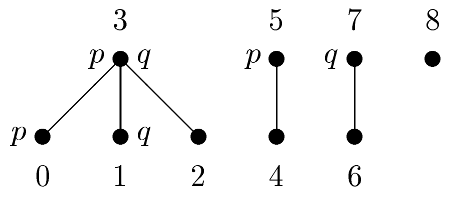
515 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$
$(((\neg p\to q)\to((p\to q)\to p))\to((\neg\neg p\to(\neg q\to p))\to p)))$ $\{2,3,5,6,7,8,9,10,11,12,13,14\}$

516 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$
$((\neg\neg p\to((p\to q)\to q))\to((\neg\neg p\to(\neg q\to p))\to p)))$ $\{2,3,5,10,11,14\}$

517 $((((\neg\neg q\to q)\to p)\to((\neg\neg q\to p)\to p))\to$
$((\neg\neg p\to((p\to q)\to q))\to((\neg\neg p\to(\neg q\to p))\to((\neg p\to\neg q)\to p))))$ $\{2,3,5,6,7,10,11,12,14\}$

B.2 The diagram of $\mathbf{H}_3^2$

The logic $\mathbf{H}_3$ was introduced in subsection 3.4.1. In the computation of the diagram of the fragment $\mathbf{H}_3^2$ the exact model of this fragment was used:



32. FIGURE. *The exact model of* $\mathbf{H}_3^2$.

Listed are the formulas in $\mathbf{H}_3^2$ and their valuations in this model.

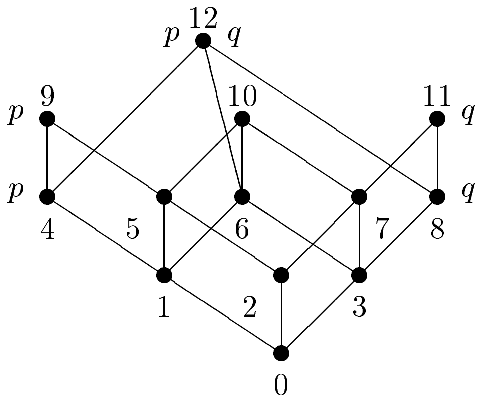
0	$(p \wedge \neg p)$	$\{\}$
1	$(p \to p)$	$\{0,1,2,3,4,5,6,7,8\}$
2	p	$\{0,3,5\}$
3	q	$\{1,3,7\}$
4	$(p \wedge q)$	$\{3\}$
5	$(p \vee q)$	$\{0,1,3,5,7\}$
6	$\neg p$	$\{6,7,8\}$
7,	$\neg q$	$\{4,5,8\}$
8	$(p \to q)$	$\{1,2,3,6,7,8\}$
9	$(q \to p)$	$\{0,2,3,4,5,8\}$
10	$\neg(p \wedge q)$	$\{4,5,6,7,8\}$
11	$\neg(p \vee q)$	$\{8\}$
12	$((p \vee q) \to (p \wedge q))$	$\{2,3,8\}$
13	$(p \vee \neg p)$	$\{0,3,5,6,7,8\}$
14	$(q \wedge \neg p)$	$\{7\}$
15	$(q \vee \neg p)$	$\{1,3,6,7,8\}$
16	$((p \wedge q) \vee \neg p)$	$\{3,6,7,8\}$
17	$((p \vee q) \vee \neg p)$	$\{0,1,3,5,6,7,8\}$
18	$(p \wedge \neg q)$	$\{5\}$
19	$(p \vee \neg q)$	$\{0,3,4,5,8\}$
20	$(q \vee \neg q)$	$\{1,3,4,5,7,8\}$
21	$((p \wedge q) \vee \neg q)$	$\{3,4,5,8\}$
22	$((p \vee q) \vee \neg q)$	$\{0,1,3,4,5,7,8\}$
23	$(p \vee (p \to q))$	$\{0,1,2,3,5,6,7,8\}$
24	$(\neg q \vee (p \to q))$	$\{1,2,3,4,5,6,7,8\}$
25	$(q \vee (q \to p))$	$\{0,1,2,3,4,5,7,8\}$
26	$(\neg p \vee (q \to p))$	$\{0,2,3,4,5,6,7,8\}$
27	$(p \vee \neg(p \wedge q))$	$\{0,3,4,5,6,7,8\}$
28	$(q \vee \neg(p \wedge q))$	$\{1,3,4,5,6,7,8\}$
29	$((p \wedge q) \vee \neg(p \wedge q))$	$\{3,4,5,6,7,8\}$
30	$((p \vee q) \wedge \neg(p \wedge q))$	$\{5,7\}$
31	$((p \vee q) \vee \neg(p \wedge q))$	$\{0,1,3,4,5,6,7,8\}$
32	$(p \vee \neg(p \vee q))$	$\{0,3,5,8\}$
33	$(q \vee \neg(p \vee q))$	$\{1,3,7,8\}$
34	$((p \wedge q) \vee \neg(p \vee q))$	$\{3,8\}$
35	$((p \vee q) \vee \neg(p \vee q))$	$\{0,1,3,5,7,8\}$
36	$(p \vee ((p \vee q) \to (p \wedge q)))$	$\{0,2,3,5,8\}$
37	$(q \vee ((p \vee q) \to (p \wedge q)))$	$\{1,2,3,7,8\}$
38	$((p \vee q) \vee ((p \vee q) \to (p \wedge q)))$	$\{0,1,2,3,5,7,8\}$
39	$(\neg p \vee ((p \vee q) \to (p \wedge q)))$	$\{2,3,6,7,8\}$
40	$(\neg q \vee ((p \vee q) \to (p \wedge q)))$	$\{2,3,4,5,8\}$
41	$(\neg(p \wedge q) \vee ((p \vee q) \to (p \wedge q)))$	$\{2,3,4,5,6,7,8\}$
42	$(q \wedge (p \vee \neg p))$	$\{3,7\}$
43	$((p \vee q) \wedge (p \vee \neg p))$	$\{0,3,5,7\}$
44	$(\neg q \wedge (p \vee \neg p))$	$\{5,8\}$
45	$(\neg(p \wedge q) \wedge (p \vee \neg p))$	$\{5,6,7,8\}$

46	$(((p \vee q)\rightarrow(p \wedge q)) \vee (p \vee \neg p))$	$\{0,2,3,5,6,7,8\}$
47	$(\neg q \vee (q \wedge \neg p))$	$\{4,5,7,8\}$
48	$((q\rightarrow p) \vee (q \wedge \neg p))$	$\{0,2,3,4,5,7,8\}$
49	$(\neg(p \vee q) \vee (q \wedge \neg p))$	$\{7,8\}$
50	$(((p \vee q)\rightarrow(p \wedge q)) \vee (q \wedge \neg p))$	$\{2,3,7,8\}$
51	$(q \vee (p \wedge \neg q))$	$\{1,3,5,7\}$
52	$((p \wedge q) \vee (p \wedge \neg q))$	$\{3,5\}$
53	$((p\rightarrow q) \vee (p \wedge \neg q))$	$\{1,2,3,5,6,7,8\}$
54	$(((p \vee q)\rightarrow(p \wedge q)) \vee (p \wedge \neg q))$	$\{2,3,5,8\}$
55	$((q \vee \neg p) \vee (p \wedge \neg q))$	$\{1,3,5,6,7,8\}$
56	$(((p \wedge q) \vee \neg p) \vee (p \wedge \neg q))$	$\{3,5,6,7,8\}$
57	$((q \wedge \neg p) \vee (p \vee \neg q))$	$\{0,3,4,5,7,8\}$
58	$(((p \vee q)\rightarrow(p \wedge q)) \vee (q \vee \neg q))$	$\{1,2,3,4,5,7,8\}$
59	$((p \vee \neg p) \wedge (q \vee \neg q))$	$\{3,5,7,8\}$
60	$(((p \wedge q) \vee \neg p) \wedge (q \vee \neg q))$	$\{3,7,8\}$
61	$(((p \vee q) \vee \neg p) \wedge (q \vee \neg q))$	$\{1,3,5,7,8\}$
62	$((p \vee \neg p) \wedge ((p \wedge q) \vee \neg q))$	$\{3,5,8\}$
63	$((q \wedge \neg p) \vee ((p \wedge q) \vee \neg q))$	$\{3,4,5,7,8\}$
64	$((p \vee \neg p) \wedge ((p \vee q) \vee \neg q))$	$\{0,3,5,7,8\}$
65	$((p \vee q) \wedge ((p \wedge q) \vee \neg(p \wedge q)))$	$\{3,5,7\}$
66	$(\neg(p \vee q) \vee ((p \vee q) \wedge \neg(p \wedge q)))$	$\{5,7,8\}$
67	$(((p \vee q)\rightarrow(p \wedge q)) \vee ((p \vee q) \wedge \neg(p \wedge q)))$	$\{2,3,5,7,8\}$
68	$((q \wedge \neg p) \vee (p \vee ((p \vee q)\rightarrow(p \wedge q))))$	$\{0,2,3,5,7,8\}$
69	$((p \wedge \neg q) \vee (q \vee ((p \vee q)\rightarrow(p \wedge q))))$	$\{1,2,3,5,7,8\}$
$7,0$	$((p \wedge \neg q) \vee (\neg p \vee ((p \vee q)\rightarrow(p \wedge q))))$	$\{2,3,5,6,7,8\}$
$7,1$	$((q \wedge \neg p) \vee (\neg q \vee ((p \vee q)\rightarrow(p \wedge q))))$	$\{2,3,4,5,7,8\}$
$7,2$	$\neg\neg p$	$\{0,1,2,3,4,5\}$
$7,3$	$(\neg p\rightarrow q)$	$\{0,1,2,3,4,5,7\}$
$7,4$	$\neg\neg q$	$\{0,1,2,3,6,7\}$
$7,5$	$(\neg q\rightarrow p)$	$\{0,1,2,3,5,6,7\}$
$7,6$	$(\neg p\rightarrow\neg q)$	$\{0,1,2,3,4,5,8\}$
$7,7$	$(\neg q\rightarrow\neg p)$	$\{0,1,2,3,6,7,8\}$
$7,8$	$\neg(p\rightarrow q)$	$\{4,5\}$
$7,9$	$((p\rightarrow q)\rightarrow p)$	$\{0,3,4,5\}$
80	$((p\rightarrow q)\rightarrow q)$	$\{0,1,3,4,5,7\}$
81	$\neg(q\rightarrow p)$	$\{6,7\}$
82	$((q\rightarrow p)\rightarrow p)$	$\{0,1,3,5,6,7\}$
83	$((q\rightarrow p)\rightarrow q)$	$\{1,3,6,7\}$
84	$\neg\neg(p \wedge q)$	$\{0,1,2,3\}$
85	$(\neg(p \wedge q)\rightarrow p)$	$\{0,1,2,3,5\}$
86	$(\neg(p \wedge q)\rightarrow q)$	$\{0,1,2,3,7\}$
87	$(\neg(p \wedge q)\rightarrow(p \vee q))$	$\{0,1,2,3,5,7\}$
88	$\neg\neg(p \vee q)$	$\{0,1,2,3,4,5,6,7\}$
89	$(\neg(p \wedge q)\rightarrow\neg(p \vee q))$	$\{0,1,2,3,8\}$
90	$\neg((p \vee q)\rightarrow(p \wedge q))$	$\{4,5,6,7\}$
91	$(((p \vee q)\rightarrow(p \wedge q))\rightarrow p)$	$\{0,1,3,4,5,6,7\}$
92	$((p \vee \neg p)\rightarrow q)$	$\{1,2,3,7\}$
93	$((p \vee \neg p)\rightarrow(p \wedge q))$	$\{1,2,3\}$
94	$((p \vee \neg p)\rightarrow((p \vee q)\rightarrow(p \wedge q)))$	$\{1,2,3,8\}$
95	$((p\rightarrow q)\rightarrow(q \wedge \neg p))$	$\{4,5,7\}$
96	$((q \vee \neg p)\rightarrow p)$	$\{0,2,3,4,5\}$
97	$(((p \vee q) \vee \neg p)\rightarrow(p \wedge q))$	$\{2,3\}$
98	$((q\rightarrow p)\rightarrow(p \wedge \neg q))$	$\{5,6,7\}$
99	$((p \vee \neg q)\rightarrow q)$	$\{1,2,3,6,7\}$
100	$((q \vee \neg q)\rightarrow p)$	$\{0,2,3,5\}$
101	$((q \vee \neg q)\rightarrow(p \wedge q))$	$\{0,2,3\}$
102	$((q \vee \neg q)\rightarrow((p \vee q)\rightarrow(p \wedge q)))$	$\{0,2,3,8\}$
103	$((q \vee \neg q)\rightarrow((p \wedge q) \vee \neg p))$	$\{0,2,3,6,7,8\}$
104	$((p \vee \neg p)\rightarrow((p \wedge q) \vee \neg q))$	$\{1,2,3,4,5,8\}$
105	$((\neg q \vee (p\rightarrow q))\rightarrow q)$	$\{0,1,3,7\}$
106	$((\neg q \vee (p\rightarrow q))\rightarrow(p \wedge q))$	$\{0,3\}$
107	$((\neg q \vee (p\rightarrow q))\rightarrow(q \vee \neg p))$	$\{0,1,3,6,7,8\}$
108	$((\neg q \vee (p\rightarrow q))\rightarrow((p \wedge q) \vee \neg p))$	$\{0,3,6,7,8\}$
109	$((\neg p \vee (q\rightarrow p))\rightarrow p)$	$\{0,1,3,5\}$
110	$((\neg p \vee (q\rightarrow p))\rightarrow(p \wedge q))$	$\{1,3\}$
111	$((\neg p \vee (q\rightarrow p))\rightarrow(p \vee \neg q))$	$\{0,1,3,4,5,8\}$
112	$((\neg p \vee (q\rightarrow p))\rightarrow((p \wedge q) \vee \neg q))$	$\{1,3,4,5,8\}$
113	$(\neg(p \wedge q)\rightarrow(p \vee \neg(p \vee q)))$	$\{0,1,2,3,5,8\}$

114	$((\neg p \lor (q\to p))\to(p \lor \neg(p \lor q)))$	$\{0,1,3,5,8\}$
115	$(\neg(p \land q)\to(q \lor \neg(p \lor q)))$	$\{0,1,2,3,7,8\}$
116	$((\neg q \lor (p\to q))\to(q \lor \neg(p \lor q)))$	$\{0,1,3,7,8\}$
117	$((\neg q \lor (p\to q))\to((p \land q) \lor \neg(p \lor q)))$	$\{0,3,8\}$
118	$((\neg p \lor (q\to p))\to((p \land q) \lor \neg(p \lor q)))$	$\{1,3,8\}$
119	$((\neg p \lor ((p \lor q)\to(p \land q)))\to p)$	$\{0,1,3,4,5\}$
120	$((\neg q \lor ((p \lor q)\to(p \land q)))\to q)$	$\{0,1,3,6,7\}$
121	$((\neg(p \land q) \lor ((p \lor q)\to(p \land q)))\to(p \land q))$	$\{0,1,3\}$
122	$((\neg(p \land q) \lor ((p \lor q)\to(p \land q)))\to((p \land q) \lor \neg(p \lor q)))$	$\{0,1,3,8\}$
123	$((p\to q)\to(q \land (p \lor \neg p)))$	$\{0,3,4,5,7\}$
124	$((q \lor \neg p)\to(q \land (p \lor \neg p)))$	$\{0,2,3,4,5,7\}$
125	$(((p \lor q) \lor \neg p)\to(q \land (p \lor \neg p)))$	$\{2,3,7\}$
126	$((q \lor \neg q)\to(q \land (p \lor \neg p)))$	$\{0,2,3,6,7\}$
127	$(((p \lor q) \lor \neg q)\to(q \land (p \lor \neg p)))$	$\{2,3,6,7\}$
128	$((\neg q \lor (p\to q))\to(q \land (p \lor \neg p)))$	$\{0,3,7\}$
129	$((q \lor (q\to p))\to(q \land (p \lor \neg p)))$	$\{3,6,7\}$
130	$((q \lor \neg(p \land q))\to(q \land (p \lor \neg p)))$	$\{0,2,3,7\}$
131	$((q \lor \neg(p \lor q))\to(q \land (p \lor \neg p)))$	$\{0,2,3,4,5,6,7\}$
132	$((q \lor ((p \lor q)\to(p \land q)))\to(q \land (p \lor \neg p)))$	$\{0,3,4,5,6,7\}$
133	$((q \lor \neg q)\to((p \lor q) \land (p \lor \neg p)))$	$\{0,2,3,5,6,7\}$
134	$((q \lor (q\to p))\to((p \lor q) \land (p \lor \neg p)))$	$\{0,3,5,6,7\}$
135	$((q \lor \neg(p \land q))\to((p \lor q) \land (p \lor \neg p)))$	$\{0,2,3,5,7\}$
136	$((\neg q \lor (p\to q))\to(((p \lor q)\to(p \land q)) \lor (q \land \neg p)))$	$\{0,2,3,7,8\}$
137	$((q\to p)\to(q \lor (p \land \neg q)))$	$\{1,3,5,6,7\}$
138	$((p \lor \neg p)\to(q \lor (p \land \neg q)))$	$\{1,2,3,4,5,7\}$
139	$((p \lor \neg q)\to(q \lor (p \land \neg q)))$	$\{1,2,3,5,6,7\}$
140	$((p \lor (p\to q))\to(q \lor (p \land \neg q)))$	$\{1,3,4,5,7\}$
141	$((p \lor \neg(p \land q))\to(q \lor (p \land \neg q)))$	$\{1,2,3,5,7\}$
142	$((p \lor \neg(p \lor q))\to(q \lor (p \land \neg q)))$	$\{1,2,3,4,5,6,7\}$
143	$((p \lor ((p \lor q)\to(p \land q)))\to(q \lor (p \land \neg q)))$	$\{1,3,4,5,6,7\}$
144	$((p \lor \neg p)\to((p \land q) \lor (p \land \neg q)))$	$\{1,2,3,4,5\}$
145	$(((p \lor q) \lor \neg p)\to((p \land q) \lor (p \land \neg q)))$	$\{2,3,4,5\}$
146	$(((p \lor q) \lor \neg q)\to((p \land q) \lor (p \land \neg q)))$	$\{2,3,5\}$
147	$((p \lor (p\to q))\to((p \land q) \lor (p \land \neg q)))$	$\{3,4,5\}$
148	$((\neg p \lor (q\to p))\to((p \land q) \lor (p \land \neg q)))$	$\{1,3,5\}$
149	$((p \lor \neg(p \land q))\to((p \land q) \lor (p \land \neg q)))$	$\{1,2,3,5\}$
150	$((((p \lor q)\to(p \land q)) \lor (p \lor \neg p))\to((p \land q) \lor (p \land \neg q)))$	$\{1,3,4,5\}$
151	$((\neg p \lor (q\to p))\to(((p \lor q)\to(p \land q)) \lor (p \land \neg q)))$	$\{1,2,3,5,8\}$
152	$((((p \lor q)\to(p \land q)) \lor (q \lor \neg q))\to(q \land (p \lor \neg p)))$	$\{0,3,6,7\}$
153	$((\neg q \lor (p\to q))\to(((p \land q) \lor \neg p) \land (q \lor \neg q)))$	$\{0,3,7,8\}$
154	$((\neg p \lor (q\to p))\to((p \lor \neg p) \land ((p \land q) \lor \neg q)))$	$\{1,3,5,8\}$
155	$(((p \lor q) \lor \neg p)\to((p \lor q) \land ((p \land q) \lor \neg(p \land q))))$	$\{2,3,4,5,7\}$
156	$(((p \lor q) \lor \neg q)\to((p \lor q) \land ((p \land q) \lor \neg(p \land q))))$	$\{2,3,5,6,7\}$
157	$((p \lor (p\to q))\to((p \lor q) \land ((p \land q) \lor \neg(p \land q))))$	$\{3,4,5,7\}$
158	$((q \lor (q\to p))\to((p \lor q) \land ((p \land q) \lor \neg(p \land q))))$	$\{3,5,6,7\}$
159	$(((p \lor q) \lor \neg(p \land q))\to((p \lor q) \land ((p \land q) \lor \neg(p \land q))))$	$\{2,3,5,7\}$
160	$(((p \lor q) \lor \neg(p \lor q))\to((p \lor q) \land ((p \land q) \lor \neg(p \land q))))$	$\{2,3,4,5,6,7\}$
161	$(((p \lor q) \lor ((p \lor q)\to(p \land q)))\to((p \lor q) \land ((p \land q) \lor \neg(p \land q))))$	$\{3,4,5,6,7\}$

B.3 The diagram of the fragment $\mathbf{IpL}_1^2$

The fragment $\mathbf{IpL}_m^n$ with restricted nesting of implication was treated in Chapter 4. The diagram of $\mathbf{IpL}_1^2$, listed below, was computed using the exact Kripke model of the fragment (compare figure 26):



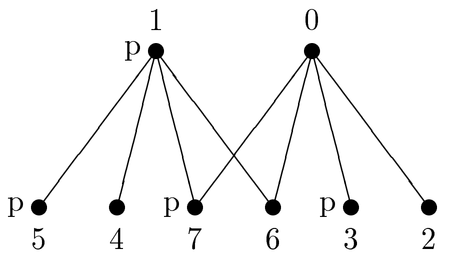
33. FIGURE. *The exact model of* $\mathbf{IpL}_1^2$.

	mu = 0	
2	p	$\{4, 9, 12\}$
3	q	$\{8, 11, 12\}$
4	$(p \wedge q)$	$\{12\}$
5	$(p \vee q)$	$\{4, 8, 9, 11, 12\}$
	mu = 1	
0	$(p \wedge \neg p)$	$\{\}$
1	$(p \to p)$	$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$
6	$\neg p$	$\{7, 10, 11\}$
7	$\neg q$	$\{5, 9, 10\}$
8	$(p \to q)$	$\{3, 6, 7, 8, 10, 11, 12\}$
9	$(q \to p)$	$\{1, 4, 5, 6, 9, 10, 12\}$
10	$\neg(p \wedge q)$	$\{2, 5, 7, 9, 10, 11\}$
11	$\neg(p \vee q)$	$\{10\}$
12	$((p \vee q) \to (p \wedge q))$	$\{6, 10, 12\}$
13	$(p \vee \neg p)$	$\{4, 7, 9, 10, 11, 12\}$
14	$(q \wedge \neg p)$	$\{11\}$
15	$(q \vee \neg p)$	$\{7, 8, 10, 11, 12\}$
16	$((p \wedge q) \vee \neg p)$	$\{7, 10, 11, 12\}$
17	$((p \vee q) \vee \neg p)$	$\{4, 7, 8, 9, 10, 11, 12\}$
18	$(p \wedge \neg q)$	$\{9\}$
19	$(p \vee \neg q)$	$\{4, 5, 9, 10, 12\}$
20	$(q \vee \neg q)$	$\{5, 8, 9, 10, 11, 12\}$
21	$((p \wedge q) \vee \neg q)$	$\{5, 9, 10, 12\}$
22	$((p \vee q) \vee \neg q)$	$\{4, 5, 8, 9, 10, 11, 12\}$
23	$(\neg p \vee \neg q)$	$\{5, 7, 9, 10, 11\}$
24	$(p \vee (p \to q))$	$\{3, 4, 6, 7, 8, 9, 10, 11, 12\}$
25	$(\neg q \vee (p \to q))$	$\{3, 5, 6, 7, 8, 9, 10, 11, 12\}$
26	$(q \vee (q \to p))$	$\{1, 4, 5, 6, 8, 9, 10, 11, 12\}$
27	$(\neg p \vee (q \to p))$	$\{1, 4, 5, 6, 7, 9, 10, 11, 12\}$
28	$((p \to q) \vee (q \to p))$	$\{1, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$
29	$(p \vee \neg(p \wedge q))$	$\{2, 4, 5, 7, 9, 10, 11, 12\}$
30	$(q \vee \neg(p \wedge q))$	$\{2, 5, 7, 8, 9, 10, 11, 12\}$
31	$((p \wedge q) \vee \neg(p \wedge q))$	$\{2, 5, 7, 9, 10, 11, 12\}$
32	$((p \vee q) \wedge \neg(p \wedge q))$	$\{9, 11\}$
33	$((p \vee q) \vee \neg(p \wedge q))$	$\{2, 4, 5, 7, 8, 9, 10, 11, 12\}$

34	$((p \to q) \vee \neg(p \wedge q))$	$\{2,3,5,6,7,8,9,10,11,12\}$
35	$((q \to p) \vee \neg(p \wedge q))$	$\{1,2,4,5,6,7,9,10,11,12\}$
36	$(p \vee \neg(p \vee q))$	$\{4,9,10,12\}$
37	$(q \vee \neg(p \vee q))$	$\{8,10,11,12\}$
38	$((p \wedge q) \vee \neg(p \vee q))$	$\{10,12\}$
39	$((p \vee q) \vee \neg(p \vee q))$	$\{4,8,9,10,11,12\}$
40	$(p \vee ((p \vee q) \to (p \wedge q)))$	$\{4,6,9,10,12\}$
41	$(q \vee ((p \vee q) \to (p \wedge q)))$	$\{6,8,10,11,12\}$
42	$((p \vee q) \vee ((p \vee q) \to (p \wedge q)))$	$\{4,6,8,9,10,11,12\}$
43	$(\neg p \vee ((p \vee q) \to (p \wedge q)))$	$\{6,7,10,11,12\}$
44	$(\neg q \vee ((p \vee q) \to (p \wedge q)))$	$\{5,6,9,10,12\}$
45	$(\neg(p \wedge q) \vee ((p \vee q) \to (p \wedge q)))$	$\{2,5,6,7,9,10,11,12\}$
46	$(q \wedge (p \vee \neg p))$	$\{11,12\}$
47	$((p \vee q) \wedge (p \vee \neg p))$	$\{4,9,11,12\}$
48	$(\neg q \wedge (p \vee \neg p))$	$\{9,10\}$
49	$(\neg q \vee (p \vee \neg p))$	$\{4,5,7,9,10,11,12\}$
50	$(\neg(p \wedge q) \wedge (p \vee \neg p))$	$\{7,9,10,11\}$
51	$(((p \vee q) \to (p \wedge q)) \vee (p \vee \neg p))$	$\{4,6,7,9,10,11,12\}$
52	$(\neg q \vee (q \wedge \neg p))$	$\{5,9,10,11\}$
53	$((q \to p) \vee (q \wedge \neg p))$	$\{1,4,5,6,9,10,11,12\}$
54	$(\neg(p \vee q) \vee (q \wedge \neg p))$	$\{10,11\}$
55	$(((p \vee q) \to (p \wedge q)) \vee (q \wedge \neg p))$	$\{6,10,11,12\}$
56	$(\neg q \vee (q \vee \neg p))$	$\{5,7,8,9,10,11,12\}$
57	$((q \to p) \vee (q \vee \neg p))$	$\{1,4,5,6,7,8,9,10,11,12\}$
58	$(((p \vee q) \to (p \wedge q)) \vee (q \vee \neg p))$	$\{6,7,8,10,11,12\}$
59	$(\neg q \vee ((p \wedge q) \vee \neg p))$	$\{5,7,9,10,11,12\}$
60	$(\neg q \vee ((p \vee q) \vee \neg p))$	$\{4,5,7,8,9,10,11,12\}$
61	$(((p \vee q) \to (p \wedge q)) \vee ((p \vee q) \vee \neg p))$	$\{4,6,7,8,9,10,11,12\}$
62	$(q \vee (p \wedge \neg q))$	$\{8,9,11,12\}$
63	$((p \wedge q) \vee (p \wedge \neg q))$	$\{9,12\}$
64	$((p \to q) \vee (p \wedge \neg q))$	$\{3,6,7,8,9,10,11,12\}$
65	$(((p \vee q) \to (p \wedge q)) \vee (p \wedge \neg q))$	$\{6,9,10,12\}$
66	$((q \vee \neg p) \vee (p \wedge \neg q))$	$\{7,8,9,10,11,12\}$
67	$(((p \wedge q) \vee \neg p) \vee (p \wedge \neg q))$	$\{7,9,10,11,12\}$
68	$((p \to q) \vee (p \vee \neg q))$	$\{3,4,5,6,7,8,9,10,11,12\}$
69	$(((p \vee q) \to (p \wedge q)) \vee (p \vee \neg q))$	$\{4,5,6,9,10,12\}$
70	$((q \wedge \neg p) \vee (p \vee \neg q))$	$\{4,5,9,10,11,12\}$
71	$(((p \vee q) \to (p \wedge q)) \vee (q \vee \neg q))$	$\{5,6,8,9,10,11,12\}$
72	$((p \vee \neg p) \wedge (q \vee \neg q))$	$\{9,10,11,12\}$
73	$(((p \wedge q) \vee \neg p) \wedge (q \vee \neg q))$	$\{10,11,12\}$
74	$(((p \vee q) \vee \neg p) \wedge (q \vee \neg q))$	$\{8,9,10,11,12\}$
75	$((p \vee \neg p) \wedge ((p \wedge q) \vee \neg q))$	$\{9,10,12\}$
76	$((q \wedge \neg p) \vee ((p \wedge q) \vee \neg q))$	$\{5,9,10,11,12\}$
77	$(((p \vee q) \to (p \wedge q)) \vee ((p \vee q) \vee \neg q))$	$\{4,5,6,8,9,10,11,12\}$
78	$((p \vee \neg p) \wedge ((p \vee q) \vee \neg q))$	$\{4,9,10,11,12\}$
79	$(((p \vee q) \to (p \wedge q)) \vee (\neg p \vee \neg q))$	$\{5,6,7,9,10,11,12\}$
80	$(\neg(p \wedge q) \vee (p \vee (p \to q)))$	$\{2,3,4,5,6,7,8,9,10,11,12\}$
81	$(\neg(p \wedge q) \vee (q \vee (q \to p)))$	$\{1,2,4,5,6,7,8,9,10,11,12\}$
82	$(\neg(p \wedge q) \vee ((p \to q) \vee (q \to p)))$	$\{1,2,3,4,5,6,7,8,9,10,11,12\}$
83	$(((p \vee q) \to (p \wedge q)) \vee (p \vee \neg(p \wedge q)))$	$\{2,4,5,6,7,9,10,11,12\}$
84	$(((p \vee q) \to (p \wedge q)) \vee (q \vee \neg(p \wedge q)))$	$\{2,5,6,7,8,9,10,11,12\}$
85	$((p \vee q) \wedge ((p \wedge q) \vee \neg(p \wedge q)))$	$\{9,11,12\}$
86	$(\neg(p \vee q) \vee ((p \vee q) \wedge \neg(p \wedge q)))$	$\{9,10,11\}$
87	$(((p \vee q) \to (p \wedge q)) \vee ((p \vee q) \wedge \neg(p \wedge q)))$	$\{6,9,10,11,12\}$
88	$(((p \vee q) \to (p \wedge q)) \vee ((p \vee q) \vee \neg(p \wedge q)))$	$\{2,4,5,6,7,8,9,10,11,12\}$
89	$((q \wedge \neg p) \vee (p \vee ((p \vee q) \to (p \wedge q))))$	$\{4,6,9,10,11,12\}$
90	$((\neg p \vee \neg q) \vee (p \vee ((p \vee q) \to (p \wedge q))))$	$\{4,5,6,7,9,10,11,12\}$
91	$((p \wedge \neg q) \vee (q \vee ((p \vee q) \to (p \wedge q))))$	$\{6,8,9,10,11,12\}$
92	$((\neg p \vee \neg q) \vee (q \vee ((p \vee q) \to (p \wedge q))))$	$\{5,6,7,8,9,10,11,12\}$
93	$((\neg p \vee \neg q) \vee ((p \vee q) \vee ((p \vee q) \to (p \wedge q))))$	$\{4,5,6,7,8,9,10,11,12\}$
94	$((p \wedge \neg q) \vee (\neg p \vee ((p \vee q) \to (p \wedge q))))$	$\{6,7,9,10,11,12\}$
95	$((q \wedge \neg p) \vee (\neg q \vee ((p \vee q) \to (p \wedge q))))$	$\{5,6,9,10,11,12\}$
96	$((\neg q \vee ((p \vee q) \to (p \wedge q))) \vee ((p \vee q) \wedge (p \vee \neg p)))$	$\{4,5,6,9,10,11,12\}$
97	$((q \vee ((p \vee q) \to (p \wedge q))) \vee (\neg(p \wedge q) \wedge (p \vee \neg p)))$	$\{6,7,8,9,10,11,12\}$

B.4 The exactly provable formulas in $\mathbf{L}_1^1$

The exactly provable formulas in $\mathbf{L}_1^1$ where computed using the exact model:



34. FIGURE. *An exact model of L_1^1.*

An explanation of the calculation of the list below can be found in section 5.5

For each formula the corresponding set of 1, 1-types is given on the right. Note that for the bracketing the priority of $\wedge$ is higher than $\vee$ and $\rightarrow$. Likewise $\vee$ has a higher priority than $\rightarrow$.

1: $p\rightarrow p$	$\{0, 1, 2, 3, 4, 5, 6, 7\}$
2: $p\rightarrow\Box\bot$	$\{0, 1, 2, 4, 6\}$
3: $p \vee \Box\bot$	$\{0, 1, 3, 5, 7\}$
4: $p\rightarrow\Box p$	$\{0, 1, 2, 4, 5, 6\}$
5: $p \wedge \Box p$	$\{1, 5\}$
6: $p \vee \Box p$	$\{0, 1, 3, 4, 5, 7\}$
7: $p\rightarrow\Box\neg p$	$\{0, 1, 2, 3, 4, 6\}$
8: $p \vee \Box\neg p$	$\{0, 1, 2, 3, 5, 7\}$
9: $\neg p \wedge \Box\neg p$	$\{0, 2\}$
10: $\Box\bot \vee \neg\Box p$	$\{0, 1, 2, 3, 6, 7\}$
11: $\Box\neg p\rightarrow\Box p$	$\{0, 1, 4, 5, 6, 7\}$
12: $p \wedge \Box p\rightarrow\Box\bot$	$\{0, 1, 2, 3, 4, 6, 7\}$
13: $p \wedge \Box\neg p\rightarrow\Box\bot$	$\{0, 1, 2, 4, 5, 6, 7\}$
14: $\Box\neg p\rightarrow p \wedge \Box\bot$	$\{1, 4, 5, 6, 7\}$
15: $\Box p\rightarrow(p \vee \Box\bot)$	$\{0, 1, 2, 3, 5, 6, 7\}$
16: $\Box\neg p\rightarrow(p \vee \Box\bot)$	$\{0, 1, 3, 4, 5, 6, 7\}$
17: $\Box p\rightarrow\neg p \wedge \Box\bot$	$\{0, 2, 3, 6, 7\}$
18: $\Box\neg p \vee (p\rightarrow\Box p)$	$\{0, 1, 2, 3, 4, 5, 6\}$
19: $\Box\neg p \vee p \wedge \neg\Box p$	$\{0, 1, 2, 3, 7\}$
20: $\Box\neg p \vee (p \vee \Box p)$	$\{0, 1, 2, 3, 4, 5, 7\}$
21: $(p \vee \Box p)\rightarrow\Box\neg p$	$\{0, 1, 2, 3, 6\}$
22: $(p \vee \Box\bot) \wedge \neg(p \wedge \Box p)$	$\{0, 3, 7\}$
23: $\Box p \vee p \wedge \neg\Box\neg p$	$\{0, 1, 4, 5, 7\}$
24: $(p \vee \Box\neg p)\rightarrow\Box p$	$\{0, 1, 4, 5, 6\}$
25: $\neg(p \wedge \Box p) \wedge (p \vee \Box\neg p)$	$\{0, 2, 3, 7\}$
26: $p \wedge \Box\bot \vee \neg(p \vee \Box\neg p)$	$\{1, 4, 6\}$
27: $p \wedge \Box p \vee \neg(p \vee \Box\neg p)$	$\{1, 4, 5, 6\}$
28: $\Box\bot \vee \neg(p \vee \Box p)$	$\{0, 1, 2, 6\}$
29: $\Box\bot \vee p \wedge \neg\Box p$	$\{0, 1, 3, 7\}$
30: $p \wedge (\Box p \vee \Box\neg p)\rightarrow\Box\bot$	$\{0, 1, 2, 4, 6, 7\}$
31: $(\Box p \vee \Box\neg p)\rightarrow(p \vee \Box\bot)$	$\{0, 1, 3, 5, 6, 7\}$
32: $\Box\bot \vee \neg(p \vee \Box\neg p)$	$\{0, 1, 4, 6\}$
33: $\Box\bot \vee p \wedge \neg\Box\neg p$	$\{0, 1, 5, 7\}$
34: $\Box\bot \vee \neg(\Box p \vee \Box\neg p)$	$\{0, 1, 6, 7\}$
35: $(p \vee \Box p) \wedge (p \wedge \Box p\rightarrow\Box\bot)$	$\{0, 1, 3, 4, 7\}$
36: $\Box\bot \vee \neg(\Box\neg p \vee p \wedge \Box p)$	$\{0, 1, 4, 6, 7\}$
37: $(p \vee \Box\neg p) \wedge (p \wedge \Box\neg p\rightarrow\Box\bot)$	$\{0, 1, 2, 5, 7\}$
38: $\Box\bot \vee \neg(\Box p \vee p \wedge \Box\neg p)$	$\{0, 1, 2, 6, 7\}$
39: $(p \vee \Box p) \wedge (\Box\neg p\rightarrow p \wedge \Box\bot)$	$\{1, 4, 5, 7\}$
40: $p \wedge \Box\bot \vee \neg(\Box\neg p \vee p \wedge \Box p)$	$\{1, 4, 6, 7\}$
41: $(p \vee \Box p)\rightarrow\Box p \wedge (p \vee \Box\bot)$	$\{0, 1, 2, 5, 6\}$
42: $(\Box p \vee \Box\neg p)\rightarrow\Box p \wedge (p \vee \Box\bot)$	$\{0, 1, 5, 6, 7\}$

43: $\Box\bot \vee (\Box p{\to}p) \wedge \neg(p \wedge \Box\neg p)$	$\{0, 1, 2, 5, 6, 7\}$
44: $(p \vee \Box\neg p){\to}\Box\neg p \wedge (p \vee \Box\bot)$	$\{0, 1, 3, 4, 6\}$
45: $\Box\bot \vee \neg\Box p \wedge (\Box\neg p{\to}p)$	$\{0, 1, 3, 6, 7\}$
46: $\Box\bot \vee \neg(p \wedge \Box p) \wedge (\Box\neg p{\to}p)$	$\{0, 1, 3, 4, 6, 7\}$
47: $(p \vee \Box p){\to}\Box\neg p \wedge \neg(p \wedge \Box\bot)$	$\{0, 2, 3, 6\}$
48: $(\Box p \vee \Box\neg p){\to}(p \vee \Box\bot) \wedge \neg(p \wedge \Box p)$	$\{0, 3, 6, 7\}$
49: $(p \vee \Box p){\to}(\Box\neg p \vee p \wedge \Box p)$	$\{0, 1, 2, 3, 5, 6\}$
50: $(p \vee \Box\neg p){\to}(\Box p \vee p \wedge \Box\neg p)$	$\{0, 1, 3, 4, 5, 6\}$
51: $\Box\neg p \vee (p \vee \Box p) \wedge \neg(p \wedge \Box p)$	$\{0, 1, 2, 3, 4, 7\}$
52: $\Box\bot \vee p \wedge \neg(\Box p \vee \Box\neg p)$	$\{0, 1, 7\}$
53: $(p \vee \Box\neg p) \wedge (p \wedge (\Box p \vee \Box\neg p){\to}\Box\bot)$	$\{0, 1, 2, 7\}$
54: $\Box p \vee (p \vee \Box\neg p) \wedge \neg(p \wedge \Box\neg p)$	$\{0, 1, 2, 4, 5, 7\}$
55: $\Box\bot \vee \neg(\Box\neg p \vee (p \vee \Box p))$	$\{0, 1, 6\}$
56: $\Box\neg p \wedge (p \vee \Box\bot) \vee \neg(\Box\neg p \vee (p \vee \Box p))$	$\{0, 1, 3, 6\}$
57: $(p \vee \Box p) \wedge (p \wedge (\Box p \vee \Box\neg p){\to}\Box\bot)$	$\{0, 1, 4, 7\}$
58: $\Box p \wedge (p \vee \Box\bot) \vee \neg(\Box\neg p \vee (p \vee \Box p))$	$\{0, 1, 5, 6\}$
59: $(\Box\neg p \vee (p \vee \Box p)) \wedge (p \wedge (\Box p \vee \Box\neg p){\to}\Box\bot)$	$\{0, 1, 2, 4, 7\}$
60: $(p \vee \Box p) \wedge (p \wedge \Box\bot \vee \neg(\Box\neg p \vee p \wedge \Box p))$	$\{1, 4, 7\}$
61: $(\Box\neg p \vee (p \vee \Box p)){\to}(p \vee \Box\bot) \wedge (\Box p \vee \Box\neg p)$	$\{0, 1, 3, 5, 6\}$
62: $(p{\to}\Box\neg p \wedge \neg\Box\bot) \wedge ((\Box p \vee \Box\neg p){\to}(p \vee \Box\bot))$	$\{0, 3, 6\}$

Appendix C

Table of fragments in IpL

For each fragment F of **IpL** in the table below, the number of equivalence classes of F^1, F^2, F^3 and F^4 have been calculated (if possible). In some cases only a lower bound of the number of elements in the diagram could be given.

fragment	$n=1$	$n=2$	$n=3$	$n=4$
$[\wedge]$	1	3	7	15
$[\vee]$	1	3	7	15
$[\wedge, \vee]$	1	4	18	166
$[\neg]$	3	6	9	12
$[\neg\neg]$	2	4	6	8
$[\wedge, \neg]$	5	23	311	66 659
$[\vee, \neg]$	7	385	$> 2^{70}$	
$[\wedge, \vee, \neg]$	7	626	$> 2^{70}$	
$[\wedge, \neg\neg]$	2	8	26	80
$[\vee, \neg\neg]$	2	9	40	281
$[\wedge, \vee, \neg\neg]$	2	19	1 889	
$[\rightarrow]$	2	14	25 165 802	$2^{623\,662\,965\,552\,393}$ $-50\,331\,618$
$[\wedge, \rightarrow]$	2	18	623 662 965 552 330	
$[\vee, \rightarrow]$	2	∞	∞	∞
$[\wedge, \vee, \rightarrow]$	2	∞	∞	∞
$[\rightarrow, \neg]$	6	518	$3 \times 2^{2\,148} - 546$	
$[\wedge, \rightarrow, \neg]$	6	2 134	$\mathbf{G}$	
$[\vee, \rightarrow, \neg]$	∞	∞	∞	∞
$[\wedge, \vee, \rightarrow, \neg]$	∞	∞	∞	∞
$[\rightarrow, \neg\neg]$	4	252	$3 \times 2^{689} - 380$	
$[\wedge, \rightarrow, \neg\neg]$	4	676	$> 2^{6\,383}$	
$[\vee, \rightarrow, \neg\neg]$	5	∞	∞	∞
$[\wedge, \vee, \rightarrow, \neg\neg]$	5	∞	∞	∞

Most of the numbers for F^1, F^2 and F^3 in the tables above can also be found in [JHR 91]. Exceptions are $|Diag([\to, \neg]^3)|$ and $Diag(|[\to, \neg\neg]^3)|$, which have been calculated by G. Renardel, and $|Diag([\wedge, \vee, \neg\neg]^3)|$ which has been computed by one of the programs of the author.

The number **G** (which approximately equals $2^{6\,385}$ and has 1 923 digits) was calculated by G. Renardel using a Mathematica program. The outcome of the program:

```
2385351090480492390853646413339133747025615299710901627960612470750032688502
8160633374326102851405827074085958557851857316972228706343515481647745510067
3005344615205148074997868754881393923444865679964852452325439433729138822091
6098391913867598073806389545947608903608155768791241781137739941904366215669
2475822999274057730123131714346501488572861062936699042596092725378572868491
2727120126756875551999208899378036731240684008111556867571467496386597453419
6639734352524036934304177304456570282321152101220432826978038593549587195612
9831811878934587983823475113519990750027976172097989085031955489803857128812
6387890256799333328705949050768971152190911269757807980550012371275543962052
0944274159250068847435305296595661994571943613671505170184414276367350905171
1680613376498354254366179877403670873176841923864088831349074573862390086523
1463501660157371767123051619018114441461150013224920279100064724918386404585
2362977616094414762999993096165017734442678516227452375506290087364604513146
8625073787337004447927030524156059024181381821779631041877411331313443793531
6573299754930440874865583477327660932604455374223117461731779709935281902018
3117687000447391980301631226240785745841846388391474813267681770574727454215
1439824203308859517469910490858730437804621971785778601804184276982651560872
1303793816082124571771381482585476984496988963204119188869627498008746051248
5777693593436069517395277231345407828098980787933234903875965383757843614426
5401046170222543436682016112844965439494799065532425819749482642048348803493
7460587870807503895520346988328053802689058378517830839398571718840621183909
0144108267526149335250474209390709830483545863841099431401775305058158120254
3786977979384559899009623073296359349660827336458692814406474316987374943103
9062915485436296897652965753764719044224517143990072975037663714964421877031
3408448192554003023114040356718230666227161668433208906675129533929667223944
3402484581279801991756
```

Bibliography

[BKK 80] J. Barwise, H.J. Keisler, K. Kunen (eds.), *The Kleene Symposium*, North-Holland, Amsterdam (1980).

[Bellissima 84] F. Bellissima, 'Atoms in modal algebras', *Zeitschr. f. math. Logik und Grundlagen d. Math.*, **30**, 303–312 (1984).

[Benthem 83] J.F.A.K. van Benthem, *Modal Logic and Classical Logic*, Napoli (1983).

[Bernardi 75] C. Bernardi, 'The fixed-point theorem for diagonalizable algebras', *Studia Logica*, **34**, 239–251, (1975).

[Beth 55] E.W. Beth, 'Semantic entailment and formal derivability', *Med. Kon. Ned. Akad. Wet.* Vol. **18**, no 13, Amsterdam (1955).

[Boolos 93] G. Boolos, *The Logic of Provability*, Cambridge University Press (1993).

[Bruijn 75a] N.G. de Bruijn, 'Exact finite models for minimal propositional calculus over a finite alphabet', *Technological University Eindhoven, Report* 75–WSK–02, (1975).

[Bruijn 75b] N.G. de Bruijn, 'An Algol program for deciding derivability in minimal propositional calculus with implication and conjunction propositional calculus over a three letter alphabet', *Technological University Eindhoven, Report* 75–WSK–06 (1975).

[CD 65] J.N. Crossley and M.A.E. Dummett (eds.), *Formal Systems and Recursive Functions*, North-Holland, Amsterdam (1965).

[CK 73] C.C. Chang and H.J. Keisler, *Model Theory*, Amsterdam (1973).

[Curry 77] H.B. Curry, *Foundations of Mathematical Logic*, second edition, New York (1977).

[Diego 66] A. Diego, *Sur les Algèbres de Hilbert*, Gauthier-Vilars, Paris (1966).

[Doets 87] K. Doets, *Completeness and Definability*, Ph.D. Thesis, University of Amsterdam (1987).

[DP 90] B.A. Davey and H.A. Priestly, *Introduction to Lattices and Order*, Cambridge University Press, Cambridge (1990).

[Ershov 89] Yu.L. Ershov (ed.), *Matematicheskaya Logika i Algoritmicheskie*

Problemy, Trudy Instituta Matematiki, vol. 12, Nauka, Novosibirsk, 120–138 (1989).

[Fine 74] K. Fine, *Logics containing K4. Part I*, J.S.L. **39**, 1, 31–42, (1974).

[Fine 85] K. Fine, *Logics containing K4. Part II*, J.S.L. **50**, 3, 619–651, (1985).

[Fitting 69] M.C. Fitting, *Intuitionistic Logic, Model Theory and Forcing*, North-Holland, Amsterdam (1969).

[Gabbay 81] D.M. Gabbay, *Semantical Investigations in Heyting's intuitionistic Logic*, Reidel, Dordrecht (1981).

[GG 90] Z. Gleit and W. Goldfarb, 'Characters and fixed points in provability logic', *Notre Dame Journal of Formal Logic* **31**, 26–36 (1990).

[Gödel 32] K. Gödel, 'Zum intuitionistischen Aussagenkalkül', *Anzeiger der Akademie der Wissenschaften in Wien*, **69**, 65–66 (1932).

[Grigolia 83] R.Sh. Grigolia, 'Finitely generated free Magari algebras' (in Russian), *Logiko-Metodologicheskie Issledovaniya*, Metsniereba, Tbilisi, 135–149 (1983).

[HC 84] G.E. Hughes and M.J. Cresswell, *A Companion to Modal Logic*, Methuen, London (1984).

[Hendriks 80] L. Hendriks, *Logische automaten, beslissen en bewijzen met de computer*, Master's Thesis (in Dutch), Department of Mathematics, University of Amsterdam (1980).

[Hendriks 93] L. Hendriks, 'Inventory of fragments and exact models in intuitionistic propositional logic', *ILLC Prepublication Series*, ML–93–11 (1993).

[HJ 96] L. Hendriks and D.H.J. de Jongh, 'Finitely generated Magari algebras and arithmetic', in *Logic and algebra, Proceedings of the Magari memorial conference, Siena 1994*, to appear (1996).

[Hosoi 66] T. Hosoi, 'The axiomatization of the intermediate propositional systems S_n of Gödel', *Journal of the Faculty of Science of the University of Tokio*, **13**, 183–187 (1966).

[HV 91] J. Halpern and M. Vardi, 'Model checking vs. theorem proving: a manifesto', in *Principles of Knowledge Representation and Reasoning: Proceedings of the* 2^{nd} *International Conference* (1991).

[Jankov 68] V.A. Jankov, 'Constructing a sequence of strongly independent superintuitionistic propositional calculi', *Sov. Math. Doc.* **9**, 806–807 (1966).

[JC 95] D.H.J. de Jongh and L.A. Chagrova, 'The decidability of dependency in intuitionistic propositional logic', *JSL* **60**, 498–504 (1995).

[JHR 91] D.H.J. de Jongh, L. Hendriks, G.R. Renardel de Lavalette, 'Computations in fragments of intuitionistic propositional logic', *Journal of Automated Reasoning* **7**, 537–561 (1991).

[De Jongh 68] D.H.J. de Jongh, *Investigations on the intuitionistic propositional calculus*, Ph.D. Thesis, University of Wisconsin, Madison (1968).

[De Jongh 70] D.H.J. de Jongh, 'A characterization of the intuitionistic propositional calculus' in [KMV 70], 211–217 (1970).

[De Jongh 80] D.H.J. de Jongh, 'A class of intuitionistic connectives' in: [BKK 80], 103–111 (1980).

[De Jongh 82] D.H.J. de Jongh, 'Formulas of one propositional variable in intuitionistic arithmetic', in [TD 82], 51–64 (1982).

[JT 66] D. de Jongh and A.S. Troelstra, 'On the connection of partially ordered sets with some pseudo-Boolean algebras', *Indag. Math.* **28** no. 3, 317–329 (1966).

[JV 95] D.H.J. de Jongh and A. Visser, 'Embeddings of Heyting algebras', *ILLC Research Report* ML-95-06 (1995) (revised edition of ML-93-14).

[KMV 70] A. Kino, J. Myhill, J. Vesley (eds.), *Intuitionism and Proof Theory*, North-Holland, Amsterdam (1970).

[Kisielewicz 88] A. Kisielewicz, 'A solution of Dedekind's problem on the number of isotone Boolean functions', *J. reine angew. Math.* **386**, 139–144 (1988).

[Kleene 62] S.C. Kleene, 'Disjunction and existence under implications in elementary intuitionistic formalisms', *JSL* **27**, 11–18 (1962).

[Kleitman 69] D. Kleitman, 'On Dedekind's problem: the number of monotone Boolean functions', *Proc. of the AMS* **21** 677–682 (1969).

[Kneale 75] W. Kneale and M. Kneale, *The Development of Logic*, Oxford University Press, London (1975).

[Kripke 65] S.A. Kripke, 'Semantical analysis of intuitionistic logic I', in [CD 65] 92–130 (1965).

[Mostowski 65] A. Mostowski, 'Thirty years of foundational studies', *Acta Phil. Fennica* **XVII** 1–180 (1965).

[Nishimura 60] I. Nishimura, 'On formulas of one variable in intuitionistic propositional logic', *JSL*,**25**, 327–331 (1960).

[Renardel 89] G.R. Renardel de Lavalette, 'Interpolation in fragments of intuitionistic propositional logic', *JSL* **54**, 1419–1430 (1989).

[Rieger 49] N.S. Rieger, 'On the lattice theory of Brouwerian propositional logic', *Acta Fac. Rerum Nat. Univ. Carolinae*, **189**, 3–40 (1949).

[Riemsdijk 85] H. van Riemsdijk, *Het genereren van diagrammen*, Master's Thesis (in Dutch), Department of Mathematics, University of Amsterdam (1985).

[De Rijke 93] M. de Rijke, *Extending Modal Logic*, Ph.D. Thesis, University of Amsterdam (1993).

[Rodenburg 86] P.H. Rodenburg, *Intuitionistic Correspondence Theory*, Ph.D. Thesis, University of Amsterdam (1986).

[Rybakov 89] V.V. Rybakov, 'On admissibility of inference rules in the modal system G' (in Russian), [Ershov 89] (1989).

[Shehtman 78] V.B. Shehtman, 'Rieger-Nishimura lattices', *Soviet Math. Dokl.*, **19/4**, 1014–1018 (1978).

[Shavrukov 93] V. Yu. Shavrukov, 'Subalgebras of diagonalizable algebras of theories containing arithmetic', *Dissertationes Mathematicae (Rozprawy Matematyczne)* **CCCXXIII**, Warszawa (1993).

[Skolem 13] Th. Skolem, 'Om konstitusjonen av den identiske kalkyls grupper', *Proc. Scan. Math. Con.*, Kristiania, 149–163 (1913).

[Sloane 73] N.J.A. Sloane, *A Handbook of Integer Sequences*, New York (1973).

[Smoryński 85] C. Smoryński, *Self-Reference and Modal Logic*, Universitext, Springer-Verlag, 1985.

[Smullyan 68] R. Smullyan, *First Order Logic*, Springer, New York (1968).

[Solovay 76] R. Solovay, 'Provability interpretations of modal logic', *Israel Journal of Mathematics* **25** 287–304 (1976).

[TD 82] A.S. Troelstra and D. van Dalen (eds.), *The L.E.J. Brouwer Centenary Symposium*, North-Holland, Amsterdam (1982).

[TD 88] A. S. Troelstra and D. van Dalen, *Constructivism in Mathematics, an Introduction* (two volumes), North-Holland, Amsterdam (1988).

[Thijsse 92] E.G.C. Thijsse, *Partial Logic and Knowledge Representation*, Ph.D. Thesis, Tilburg University (1992).

[Thomas 62] I. Thomas, 'Finite limitations on Dummett's *LC*', *Notre Dame Journal of Formal Logic*, **III/3**, 170–174 (1962).

[Troelstra 65] A. Troelstra, 'Finite logics', *Indag. Math.* **27**, 141–152 (1965).

[Urquhart 74] A. Urquhart, 'Implicational formulas in intuitionistic logic', *JSL* **39**, 661–664 (1974).

[Visser 84] A. Visser, 'The provability logics of recursively enumerable theories extending Peano arithmetic at arbitrary theories extending Peano arithmetic', *Journal of Philosophical Logic* **13** 97–113 (1984).

[VBJR 95] A. Visser, J.F.A.K.van Benthem, D.H.J. de Jongh, G.R. Renardel de Lavalette, 'NNIL, A study in intuitionistic propositional logic', in: *Modal Logics and Process Algebra - a bisimulation perspective*, eds. A. Ponse, M. de Rijke, Y. Venema, CSLI, Stanford, 289–326 (1995).

[Visser 85] A. Visser, 'Evaluation, provably deductive equivalence in Heyting's arithmetic of substitution instances of propositional formulas', *Logic Group Preprint Series*, 4, University of Utrecht, Utrecht (1985).

[Zambella 94] D. Zambella, 'Shavrukov's theorem on the subalgebras of diagonalizable algebras for theories containing $I\Delta_0 + EXP$', *Notre Dame Journal of Formal Logic* **35** 147–157 (1994).

[Zwanenburg 94] J. Zwanenburg, *Skeletons in intuitionistic propositional logic*, Master's Thesis, University of Groningen (1994).

List of symbols

$\wedge, \vee, \rightarrow, \neg, \Box, \Diamond$	8
$\neg\neg$	6
$\sim, \Rightarrow$	12
$\boxdot$	101
$\Box^n$	102
$\mathbf{CpL}$	8
$\mathbf{GL}$	99
$\mathbf{H}_3, \mathbf{H}_3^n$	52, 54
$\mathbf{IpL}$	4
$\mathbf{IpL^n}$	6
$\mathbf{IpL}_m^n$	93
$\mathbf{K}$	8
$\mathbf{K}_m^n$	24
$\mathbf{L}, \mathbf{L}_1^1$	6, 99
$\mathbf{L}_m^n$	101
$\mathbf{LC}$	54
$\mathbf{PA}$	6, 99
$\mathbf{PRL}$	99
$\vdash_{\mathbf{PA}}$	7
$\dot{\vdash}$	7, 99
$\vdash, \vdash_L$	8
$\dashv$	17
$\beta(\phi)$	24
$\gamma(A)$	115, 119, 125
$\phi_F(k)$	17
$\phi_{\mathbf{CpL}}^n(k)$	21
$\phi_m^n(k)$	101
ϕ_Q^n	21
ϕ^*	7
$\phi_m^n(C)$	105
$\psi^n(k)$	74, 88
$\lvert\phi\rvert$	117
$\vert$	44
$\mu(\phi)$	93
$\bot, \top$	8
$\mathbf{t}, *, \mathbf{f}$	54
$Diag(F)$	9
$\oplus$	17
$I(F)$	17
$\mathcal{P}^*(X)$	9, 17
$D(n)$	47
$\breve{R}$	9
$X^\circ, X^\bullet$	37
Γ, Δ	28
L, R	113
$\#X$	113
$Sub(X)$	113
A, X	114
$\alpha^n(\phi)$	78, 84, 92
$atom, atom^n$	9
$\delta(k)$	11
$\langle W, R\rangle$	9
$\langle W, R, atom\rangle$	9
$\langle W, \preceq, \omega\rangle$	12
$\nVdash$	10
$\models_{\mathcal{M}}, \Vdash_{\mathcal{M}}$	10

$\omega(\phi)$	12, 102
$\omega^n(T)$	102
$[\![\phi]\!]$	12, 27
ρ	132
$Ter(k)$	49
$ucv^n(\phi)$	84, 78, 92
$\uparrow, \underline{\uparrow}, \downarrow$	11
$\underline{\uparrow} k \downarrow$	105
$\Vdash$	9
$j_0(t), j_1(t)$	20
$\preceq$	12, 31
$Th_F(k)$	15
$Th^n_m(k)$	94
T^n_m	26, 101
$\tau_F(k)$	16
$\tau^n_m(k)$	94, 101
$\upharpoonright$	101
$\rightleftarrows$	19
$\rightleftarrows^n$, $\rightleftarrows^n_m$	19
E^n	47
E^n_M	73
ExK^n	29
ExL^n	101
$Exm(\mathbf{CpL}^n)$	23
$Exm(F)$	42
$Exm(\mathbf{H}^n_3)$	56
$Exm(\mathbf{IpL}^n_m)$	95
$Exm(\mathbf{L}^1_1)$	109
$Exm([\vee, \neg\neg, \bot]^n)$	66
$Exm([\wedge, \rightarrow, \neg]^n)$	71
$Exm([\wedge, \rightarrow]^n)$	86
$Exm([\wedge, \vee, \neg]^n)$	51
$K^\cap$	69
K^τ	33
$(\uparrow k)^T$	52
$G^n(K, L, \langle k, l \rangle)$	48
$\models G(K, L)$	39
$Umod([\vee, \neg]^n)$	63
$Umod([\wedge, \rightarrow, \neg\neg]^n)$	81
$Umod([\wedge, \neg\neg]^n)$	58
$Umod([\wedge, \vee, \neg\neg]^n)$	61
ΔX	74, 88
$Newatom^n(k)$	74, 88
$L \bullet R$	113
$L \mathbin{\underline{\bullet}} R$	115
$L \circ R$	118
$L \odot R$	118
$L\ (w; W)\ R$	129
$\delta(L \circ R), \delta(L \odot R)$	119
$\eta(L \bullet R)$	115, 120, 125
$\lambda(L \bullet R)$	119
$m(L \bullet R)$	120
$\mu(L \bullet R)$	115, 120, 125
$\sigma(X)$	115, 120, 125

Index

$\cap$-independent
  model, 69
  node, 69
  reduction, 69
ω-consistency, 100
$\Diamond$NW-rule, 125
τ-filter, 99
2-model, 52
m-equivalence, 25
n-bisimulation, 19
n-complete, 11, 102
n-dimensional hypercube, 48, 64, 67
n-maximal model, 57
n-minimal, 65
n-model, 9
n, m-bisimulation, 19
n, m-equivalence, 25
n, m-maximal exactly provable, 107
n, m-type, 96
n, m-type in **L**, 101
n, m-type in **K**, 26
$\mathbf{T}\Box L$-rule, 129
CpL Kripke model, 9
IpL Kripke model, 9
CpLtest, 114
Ctest, 116
IpLtest, 118
Itest, 122
K4Mtest, 133
K4test, 129
KMtest, 128
Ktest, 123
LMtest, 135
Ltest, 134
S4test, 134
Ttest, 129
mkDiag, 38

accessibility relation, 9
Aczel slash, 44
Algol68, 4
anticyclic, 11
arithmetical interpretation, 99
associated Kripke model, 121, 127, 132, 134
axioms, 8

Beth, 3, 4, 113
Birkhoff, 17
bisimulation, 18
bottom type, 46
bounded bisimulation, 19
box nesting, 24
branch, 113

canonical exact model, 28
canonical model for $\mathbf{L}^n$, 101
Chagrov, 51
character, 15
classical model, 12
closed

- sequent, 113
- subset, 9
- tableau, 114

closing sequent, 115
complete model, 12
completeness, 10
completion, 41
conservative extension, 50
consistent, 101
cycle, 11

De Bruijn, 4, 69
De Jongh, 4, 5, 36
decision procedure, 113
Dedekind, 47
depth, 11
diagonizable algebra, 99
diagram, Lindenbaum algebra, 3, 9
Diego, 4
direct successor, 9
domain, 9
downwards directed, 102
Dummett, 54

Ehrenfeucht, 26
Ehrenfeucht game, 39
- for $[\wedge,\vee,\neg]^n$, 48

enveloping type, 105
equivalency, 8
exact formula, 100
exact Kripke model, 15
- of $[\wedge,\rightarrow,\neg]^n$, 73
- of $[\wedge,\rightarrow]^n$, 87
- of $[\wedge,\vee,\neg]^n$, 51
- of $[\wedge,\vee]^n$, 46
- of $\mathbf{CpL}^n$, 23
- of $\mathbf{H}_3^n$, 56
- of $\mathbf{IpL}^1$, 36
- of $\mathbf{L}_1^1$, 109

exact model, 4, 12
- of $[\wedge,\rightarrow,\neg\neg]^n$, 82
- of $[\wedge,\neg\neg,\top]^n$, 58
- of $\mathbf{K}_1^1$, 28

exact provable formula, 100
exactly provable, 6, 7

Fine, 25, 36
fixed point theorem, 111
forcing, 9
formula tester, 113
Fraïssé, 26
fragment, 9

Gödel, 54
Gödel-sentence, 110
generated submodel, 11
global consequence, 10
Grzegorczyk, 146

Henkin method, 28
Hintikka, 3
Hosoi, 54

interderivable, 101
interior, 37
interpretable theory, 7, 100
intuitionistic Kripke model, 9
irreducible, 17
irreducible model, 23, 33

Jankov, 36
join-irreducible, 17

K4NW-rule, 130
Kamp, 4
Kanger, 3
knowledge representation, 3
Kripke, 3
Kripke completion, 41
Kripke frame, 9
Kripke model, 9
- associated Kripke model, 121, 127, 132, 134
- intuitionistic Kripke model, 9
- **CpL** Kripke model, 9
- **IpL** Kripke model, 9

Kripke model theory, 9

layered
- bisimulation, 18, 19
- fragments, 19

Lindenbaum, 3
LNW-rule, 134

local consequence, 10
locally finite, 20
logic
  CpL, 8
  $\mathbf{H}_3$, 52
  IpL, 8
  K, 8, 124
  K4, 129
  K4Grz, 29, 146
  L, 6, 100
  S4, 29, 134
  S5, 146
  T, 129
logical machine, 4
lower carrier, 78

Magari algebra, 99
maximal exactly provable formula, 105
maximal reduction, 33
modal degree, 24, 101
modal depth, 6, 24
model checking, 3
model equivalence, 18
model, generalized, 12

nesting of implication, 6, 93
new world rule, 124
Nishimura, 3
nodes, 9
normal form
  in $[\vee, \neg\neg]^n$, 65
  in $[\wedge, \neg\neg]^n$, 57
  in $[\wedge, \vee, \neg\neg]^n$, 60
  in $[\wedge, \vee, \neg]$, 51

open sequent, 115
open tableau, 114

p-morphism, 19
Pascal, 4
Peano arithmetic, 6, 99
Pierce formula, 54
poset, 9
predecessor, 9
predecessor set, 38
predicate logic, 4

proper $[\vee, \neg]^n$ node, 63
proper $[\wedge, \rightarrow, \neg\neg]^n$ model, 80
proper $[\wedge, \rightarrow]^n$ model, 85
proper $[\wedge, \vee, \neg\neg]^n$ model, 61
proper node, 58
propositional theory, 99
provability logic, 99, 100
provability predicate, 99
provable sentence, 110
pseudo-code, 116, 122, 128, 133, 135
pseudo-epimorphism, 19

realize, 21
reduced split sequent, 120, 126
reduction, 19
reflexive type, 104
refutable sentence, 110
Renardel de Lavalette, 5, 79, 84, 92, 172
Rieger, 3
Rieger-Nishimura lattice, 36
root, 9
Rosser-sentence, 110
rules, 8

semantic n, m-type
  in $\mathbf{IpL}^n_m$, 94
  in **K**, 26
semantic tableau, 4
  method, 113
semantic type, 6, 15, 16
  in $[\vee, \neg\neg]^n$, 65
  in $[\wedge, \rightarrow, \neg\neg]^n$, 80
  in $[\wedge, \rightarrow, \neg]^n$, 70
  in $[\wedge, \rightarrow]^n$, 85
  in $[\wedge, \neg\neg]^n$, 58
  in $[\wedge, \vee, \neg\neg]^n$, 61
  in $[\wedge, \vee, \neg]^n$, 49, 63
  in $[\wedge, \vee]^n$, 46
  in **CpL**, 23
  in $\mathbf{H}^n_3$, 55
  in **IpL**, 31
  in $\mathbf{IpL}^n_m$, 94
  in **K**, 29
  in **L**, 101

sequent calculus, 113
Shavrukov, 7, 99, 102
Skolem, 3
Solovay, 6, 99
soundness, 10
Sperner, 47
split sequent, 114
  of *CpLtest*, 114
  of *IpLtest*, 118
  of *K4test*, 129
  of *Ktest*, 124
starting worlds, 39
steady, 102
strong disjunction property (s.d.), 102
subfragment, 9
successor, 9

tableau, 113
tail model, 105
terminal model, 53
terminal node, 9
theorem prover, 113
theorem proving, 3
theory of a node, 15
Thomas, 54
three valued Heyting logic, 52, 54
traffic light, 4
Tromp, 5
type, 15
type formula, 15
type formula
  in $[\vee, \neg\neg]^n$, 66
  in $[\wedge, \rightarrow, \neg]^n$, 72
  in $[\wedge, \rightarrow]^n$, 87
  in $[\wedge, \neg\neg]^n$, 58
  in $[\wedge, \vee, \neg\neg]^n$, 61
  in $[\wedge, \vee, \neg]^n$, 49
  in $[\wedge, \vee]^n$, 46
  in **CpL**, 21
  in $\mathbf{H}_3^n$, 55
  in $\mathbf{IpL}_m^n$, 96
  in $\mathbf{K}^n$, 30
  in $\mathbf{K}_m^n$, 26
  in $\mathbf{L}^n$, 102
  in $\mathbf{L}_m^n$, 101

universal model, 18, 41, 43
  for $[\vee, \neg]^n$, 63
  for $[\wedge, \rightarrow, \neg\neg]^n$, 81
  for $[\wedge, \neg\neg]^n$, 58
  for $[\wedge, \vee, \neg\neg]^n$, 61
upper carrier, 78, 84, 92
upwards closed realizable, 105
Urquhart, 4

Van Riemsdijk, 4, 5

worlds, 9

Zwanenburg, 6, 96

Samenvatting

Dit proefschrift doet verslag van een onderzoek naar de semantiek van de intuïtionistische en de modale propositielogica. Dit onderzoek is voor een belangrijk deel geïnspireerd en mogelijk gemaakt door het experimenteren met computerprogramma's.

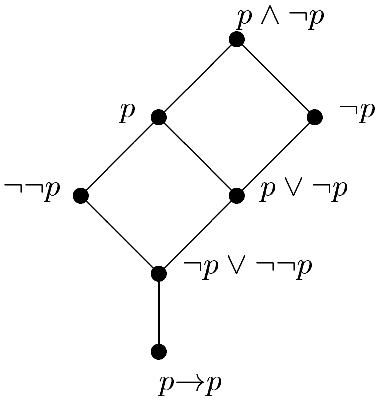
De oudste van deze computerprogramma's zijn zogenaamde *stellingtesters*, programma's waarmee kan worden uitgerekend of uit een bewering A de bewering B logisch volgt. Daarbij wordt alleen gebruik gemaakt van de *vorm* van de beweringen A en B. De computer hoeft dan geen verstand te hebben van sterrenkunde, om uit de bewering 'De Maan is niet van groene kaas' af te leiden: 'Als de Maan van groene kaas is, dan draait Venus om de Aarde'. In *Hoofdstuk 6* worden diverse programma's beschreven om, voor verschillende logische systemen, te berekenen of B uit A volgt. De belangrijkste onderdelen van deze programma's zijn opgenomen in *Appendix A*.

Door de formele taal van de propositielogica, waarin de beweringen kunnen worden geformuleerd, voldoende te beperken krijgt men een zogenaamd *fragment* waarin slechts eindig veel logisch verschillende beweringen mogelijk zijn. Voorbeelden van de beperkingen die men kan opleggen zijn het toelaten van slechts eindig veel basisbeweringen en het verbieden van een of meerdere van de connectieven (voegwoorden) uit de rij 'en' ($\wedge$), 'of' ($\vee$), 'als ... dan' ($\rightarrow$), 'niet' ($\neg$), 'mogelijk' ($\Diamond$) en 'noodzakelijk' ($\Box$). Daarbij maakt het ook nogal wat verschil welke logische afleidingsregels men in het fragment toelaat. Zo heeft $[\wedge, \vee, \rightarrow, \neg]^1_{\mathbf{CpL}}$, het fragment uit de klassieke propositielogica logica met precies één basisbewering en met als connectieven $\wedge, \vee, \rightarrow$ en $\neg$, vier echt verschillende beweringen (A, $\neg A$, $A \wedge \neg A$ en $A \rightarrow A$). Maar het fragment $[\wedge, \vee, \rightarrow, \neg]^1_{\mathbf{IpL}}$ in de intuïtionistische propositielogica, **IpL**, telt oneindig veel verschillende beweringen. Dit geldt voor alle fragmenten in **IpL** die zowel $\vee$ als $\rightarrow$ bevatten.

Als er maar eindig veel verschillende beweringen in een fragment zijn, kunnen we, in principe, alle echt verschillende beweringen uit het fragment berekenen, met behulp van een computerprogramma dat kan uitmaken of een bewering A gelijkwaardig is met de bewering B. Ook de onderlinge relaties tussen deze beweringen (wat volgt er uit wat) kunnen we op die manier in kaart brengen. Zo'n kaart van een fragment,

met daarop alle beweringen uit het fragment en hun onderlinge relaties, noemen we in dit proefschrift een *diagram.*

Hieronder is een voorbeeld van zo'n diagram getekend, in dit geval van het fragment $[\wedge, \vee, \neg]^1$ in de intuïtionistische propositielogica, met basisbewering p:



In dit voorbeeld kan het diagram nog met de hand worden berekend. Voor diagrammen met meer dan twintig beweringen is dat al haast niet meer doenlijk en moet bijvoorbeeld een beroep gedaan worden op een van de eerder genoemde stellingtesters. Uit de eerste experimenten met het berekenen van diagrammen met deze stellingtesters, eind jaren zeventig en begin jaren tachtig, bleek al snel dat zo alleen 'kleine' fragmenten (met hooguit zo'n honderd echt verschillende beweringen) in redelijke tijd in kaart te brengen zijn.

Exacte modellen

Gelukkig bestaat er ook een alternatief voor de stellingtesters, namelijk programma's die gebruik maken van *exacte Kripke-modellen.* Kripke-modellen zijn in de intuïtionistische en modale logica bekende hulpmiddelen om bijvoorbeeld situaties (en hun onderlinge relaties) mee te beschrijven waarin een bepaalde bewering A geldt en de bewering B juist niet. Dat geeft dan een tegenvoorbeeld tegen de bewering dat B uit A volgt.

Een exact Kripke-model van een fragment beschrijft precies alle tegenvoorbeelden die we nodig hebben om voor een fragment uit te maken voor welke beweringen geldt dat B uit A volgt. Elke bewering uit het fragment heeft in het exacte Kripke-model een gebied waar deze bewering geldig is. Als het gebied waar A geldig bevat is in het gebied waar B geldt, dan is B blijkbaar een logisch gevolg van A.

Het berekenen van diagrammen van fragmenten met behulp van exacte modellen gaat vele malen sneller dan met behulp van de eerder genoemde stellingtesters. Lang niet alle fragmenten hebben echter een exact Kripke-model (de situatie in **IpL** is weergegeven in figuur 1 in hoofdstuk 1). Daar staat tegenover dat we veel fragmenten kunnen beschouwen als onderdeel van een fragment dat wel een exact model heeft. Voorbeelden van de berekeningen van diagrammen met behulp van exacte modellen zijn opgenomen in *Appendix B.*

Hoofdstuk 3 van dit proefschrift is gewijd aan de berekening van de diagrammen van de eindige fragmenten in de intuïtionistische propositie logica. Daarbij wordt niet alleen gebruik gemaakt van exacte Kripke-modellen, bij de fragmenten die zich daarvoor lenen wordt ook aangegeven hoe deze exacte modellen kunnen worden geconstrueerd. Zoals uit de tabel in *Appendix C* blijkt worden de diagrammen van eindige fragmenten van **IpL** al bij een klein aantal basisbeweringen in het algemeen al snel astronomisch groot. Het werkelijk laten berekenen van de formules die bij de verschillende beweringen uit de fragmenten horen is in dat geval praktisch uitgesloten en het inzicht in de structuur van de exacte Kripke-modellen is dan vooral van theoretisch belang.

In de modale logica levert, ook met een eindig aantal basisbeweringen, het beperken van de gebruikte voegwoorden in het algemeen nog geen eindige fragmenten op. Een bekende ingreep om toch te komen tot eindige diagrammen is het beperken van de mate waarin het 'mogelijk' en 'noodzakelijk' in een bewering gestapeld voorkomen. Bij een grens van één zou bijvoorbeeld de bewering $\Box\Box A$ (het is noodzakelijk dat het noodzakelijk is dat A') niet meer tot het fragment horen.

In *Hoofdstuk 4* van dit proefschrift wordt iets soortgelijks gedaan voor de intuïtionistische propositielogica. Door het beperken van de stapeling van $\rightarrow$ leidt het samenspel van 'of' ($\vee$) en 'als ... dan' ($\rightarrow$) ook in **IpL** niet langer tot oneindig veel verschillende beweringen. Aangetoond wordt hoe voor deze fragmenten met beperkte stapeling van de implicatie exacte Kripke-modellen geconstrueerd kunnen worden.

Semantische typen

Om de exacte modellen voor fragmenten van propositielogica's te kunnen berekenen is nader onderzocht welke situaties en relaties nodig zijn om alle gewenste tegenvoorbeelden in een Kripke-model te kunnen weergeven. Wat maakt, met andere woorden, een bewering geldig in een bepaalde situatie in een Kripke-model? Het antwoord op deze vraag hangt af van de logica en van het fragment binnen die logica waarmee we werken. In het algemeen kunnen we een volledig beeld geven van een situatie met behulp van een opsomming van de basisbeweringen die er gelden, samen met een overzicht van de andere situaties die vanuit deze situatie 'denkbaar' zijn[1]. De combinatie van deze opsommingen noemen we een *semantisch type*.

Situaties die voor een bepaald fragment van een propositielogica hetzelfde semantische type hebben, gedragen zich logisch gezien eender en er gelden dezelfde beweringen uit het fragment. Het opsporen van de semantische typen voor een bepaald fragment blijkt een heel geschikte methode om een Kripke-model te maken waarin alle voor een fragment nodige tegenvoorbeelden voorhanden zijn. Vaak is zo'n model te groot om een mooi exact Kripke-model te zijn, maar als basis voor een computerprogramma om een diagram mee te berekenen voldoet het prima.

[1]Wat 'denkbaar' is, welke relaties de situaties in een model kunnen hebben, hangt van de logica in kwestie af.

In *Hoofdstuk 2* van dit proefschrift wordt de theorie over de semantische typen uiteengezet en in verband gebracht met een aantal reeds bekende resultaten over modellen en beweringen uit de klassieke, de intuïtionistische en de modale propositielogica.

Formele rekenkunde

In *Hoofdstuk 5* van het proefschrift wordt de theorie van de semantische typen toegepast op een probleem uit de formele rekenkunde, de Peano-rekenkunde **PA**. In de rekenkundige taal zelf kunnen we de bewering formuleren dat een rekenkundige zin bewijsbaar is. Als A een rekenkundige bewering is, dan wordt de rekenkundige bewering 'bewijsbaar A' ook wel geschreven als $\Box A$. De regels die voor deze vorm van 'bewijsbaarheid' gelden vormen een bijzondere modale propositielogica, de bewijsbaarheidslogica **L**.

Nemen we voor een basisbewering p in de bewijsbaarheidslogica een bepaalde rekenkundige zin (bijvoorbeeld '7 heeft 64 verschillende delers'), dan noemen we de verzameling beweringen die we kunnen maken in het fragment van **L** met één basisbewering en die geldig zijn in de rekenkunde als we voor de basisbewering een rekenkundige zin nemen, de $\mathbf{L}^1$-*theorie* van die rekenkundige zin.

Een $\mathbf{L}^1$-theorie heeft als axioma de bewering A, als A zelf een bewering uit de theorie is en alle andere beweringen in de theorie logische gevolgen zijn van A.

Zelfs bij een beperking van het fragment van **L** waarbij alleen beweringen worden toelaten waarin $\Box$ maar één keer gestapeld mag voorkomen (de stapelgrens in dit fragment is dus 2), was tot voor kort niet bekend hoeveel verschillende axioma's voor $\mathbf{L}^1_2$-theorieën er zijn.

Zoals in *Hoofdstuk 5* wordt aangetoond (en uiteindelijk met de computer kon worden berekend) zijn er precies 62 verschillende axioma's voor dit soort theorieën.

Net als bij het berekenen van het aantal verschillende beweringen in de eindige fragmenten van **IpL** is zo'n getal als uitkomst uiteindelijk niet het belangrijkste. Wat telt is dat we zoveel inzicht hebben gekregen in de structuur van fragmenten van propositielogica's dat we computerprogramma's kunnen maken om dergelijke berekeningen uit te voeren.

Titles in the ILLC Dissertation Series:

ILLC DS-1993-1: **Paul Dekker**
Transsentential Meditations; Ups and downs in dynamic semantics

ILLC DS-1993-2: **Harry Buhrman**
Resource Bounded Reductions

ILLC DS-1993-3: **Rineke Verbrugge**
Efficient Metamathematics

ILLC DS-1993-4: **Maarten de Rijke**
Extending Modal Logic

ILLC DS-1993-5: **Herman Hendriks**
Studied Flexibility

ILLC DS-1993-6: **John Tromp**
Aspects of Algorithms and Complexity

ILLC DS-1994-1: **Harold Schellinx**
The Noble Art of Linear Decorating

ILLC DS-1994-2: **Jan Willem Cornelis Koorn**
Generating Uniform User-Interfaces for Interactive Programming Environments

ILLC DS-1994-3: **Nicoline Johanna Drost**
Process Theory and Equation Solving

ILLC DS-1994-4: **Jan Jaspars**
Calculi for Constructive Communication, a Study of the Dynamics of Partial States

ILLC DS-1994-5: **Arie van Deursen**
Executable Language Definitions, Case Studies and Origin Tracking Techniques

ILLC DS-1994-6: **Domenico Zambella**
Chapters on Bounded Arithmetic & on Provability Logic

ILLC DS-1994-7: **V. Yu. Shavrukov**
Adventures in Diagonalizable Algebras

ILLC DS-1994-8: **Makoto Kanazawa**
Learnable Classes of Categorial Grammars

ILLC DS-1994-9: **Wan Fokkink**
Clocks, Trees and Stars in Process Theory

ILLC DS-1994-10: **Zhisheng Huang**
Logics for Agents with Bounded Rationality

ILLC DS-1995-1: **Jacob Brunekreef**
On Modular Algebraic Protocol Specification

ILLC DS-1995-2: **Andreja Prijatelj**
Investigating Bounded Contraction

ILLC DS-1995-3: **Maarten Marx**
Algebraic Relativization and Arrow Logic

ILLC DS-1995-4: **Dejuan Wang**
Study on the Formal Semantics of Pictures

ILLC DS-1995-5: **Frank Tip**
Generation of Program Analysis Tools

ILLC DS-1995-6: **Jos van Wamel**
Verification Techniques for Elementary Data Types and Retransmission Protocols

ILLC DS-1995-7: **Sandro Etalle**
Transformation and Analysis of (Constraint) Logic Programs

ILLC DS-1995-8: **Natasha Kurtonina**
Frames and Labels. A Modal Analysis of Categorial Inference

ILLC DS-1995-9: **G.J. Veltink**
Tools for PSF

ILLC DS-1995-10: **Giovanna Ceparello**
(to be announced)

ILLC DS-1995-11: **W.P.M. Meyer Viol**
Instantial Logic. An Investigation into Reasoning with Instances

ILLC DS-1995-12: **Szabolcs Mikulás**
Taming Logics

ILLC DS-1995-13: **Marianne Kalsbeek**
Metalogics for Logic Programming

ILLC DS-1995-14: **Rens Bod**
Enriching Linguistics with Statistics: Performance Models of Natural Language

ILLC DS-1995-15: **Marten Trautwein**
Computational Pitfalls in Tractable Grammatical Formalisms

ILLC DS-1995-16: **Sophie Fischer**
The Solution Sets of Local Search Problems

ILLC DS-1995-17: **Michiel Leezenberg**
Contexts of Metaphor

ILLC DS-1995-18: **Willem Groeneveld**
Logical Investigations into Dynamic Semantics

ILLC DS-1995-19: **Erik Aarts**
Investigations in Logic, Language and Computation

ILLC DS-1995-20: **Natasha Alechina**
Modal Quantifiers

ILLC DS-1996-1: **Lex Hendriks**
Computations in Propositional Logic