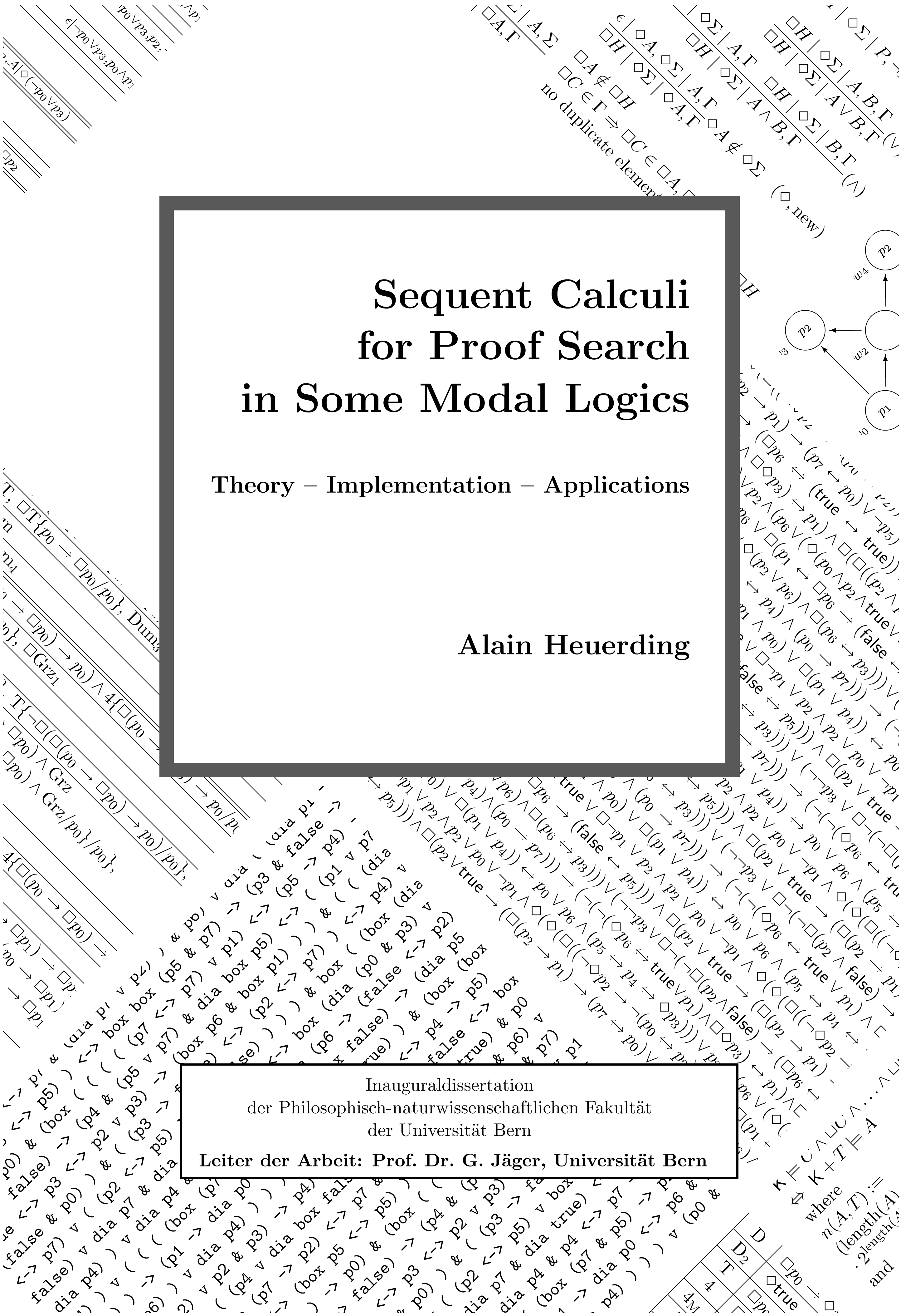


Sequent Calculi for Proof Search in Some Modal Logics

Theory – Implementation – Applications

Alain Heuerding

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

Leiter der Arbeit: Prof. Dr. G. Jäger, Universität Bern

Sequent Calculi for Proof Search in Some Modal Logics

Theory – Implementation – Applications

Alain Heuerding

Sequent Calculi for Proof Search in Some Modal Logics

Theory – Implementation – Applications

Alain Heuerding

Inauguraldissertation
der Philosophisch-naturwissenschaftlichen Fakultät
der Universität Bern

Leiter der Arbeit:
Prof. Dr. G. Jäger, IAM, Universität Bern

If you want a copy of your own, then send a request (with your address) to:

A. Heuerding
Cäcilienstr. 5
3007 Bern
Switzerland

If this address is no longer valid, then you can try to send it to anybody with surname 'Heuerding' in Switzerland.

CONTENTS

Certainly modal logic also suffers from an over-developed formalism. The literature on many-valued systems is ridiculously large, but I am sure it does not match in waste of time that on modalities.

D. Scott. Does Many-Valued Logic Have Any Use?

INTRODUCTION

The White Rabbit put on his spectacles. “Where shall I begin, please your Majesty?” he asked. “Begin at the beginning,” the King said gravely, “and go on till you come to the end: then stop.”

L. Carroll. Alice’s Adventures in Wonderland.

THE SUBJECT

Modal logics have a long tradition in logic. One way to look at a modal logic is to consider it as a definition of a set of modal formulas. The natural question that arises is whether or not a given formula is in this set, i.e. whether or not it is valid. Checking the validity by hand is a very tiresome and error-prone task even in the case of shorter formulas. Here automated proof search can help with implementations of decision procedures, which are several magnitudes faster.

In this thesis we investigate proof search based on sequent calculi for some propositional modal logics, namely for K, KT, $\mathsf{S4}$ (with and without theories) and for the tense logic K_t. One advantage of such procedures is that they are relatively easy to describe and to implement, and the LWB (Logics Workbench) shows that implementations can be reasonably fast. Note that we do not try to give one generic formalism for many logics at once, but deal separately with each of the logics. This allows us to develop calculi that make use of the properties of each logic, and we can ensure the termination of backward proof search without becoming inefficient.

Besides presenting sequent calculi for backward proof search, we have three further aims. Many results in this thesis have been known for a long time. We hope that by using a uniform presentation and notation we can make clear the interdependencies between semantics, sequent calculi, complexity results, ... This could make it worthwhile to read the thesis even for an expert or for somebody whose main interest is not sequent calculi. Secondly, we define tests and benchmarks for proof search in these logics, which make it possible to compare theorem provers and help to develop new ones. Finally, we show how efficient decision procedures can be used to solve problems in modal logic with brute force methods.

THE IDEAL READER

We start at the beginning and introduce all notions we shall use, often together with an example. Nevertheless, it is helpful to have some basic knowledge of modal logic. For readers without any foreknowledge, looking at an introduction in modal logic (for example [CZ97], [Che80], [BS86], [Rau79]) will help to understand the motivation and intuition behind the definitions and theorems. The situation is about the same in the case of sequent calculi and proof search. Although in that respect, the thesis is self-contained, but a certain understanding will make reading much easier. Ideally the reader is familiar with the usual sequent calculus for classical propositional logic, has heard something about proof search in this calculus and has tried to find proofs for some formulas by hand. The only chapter where we really presuppose knowledge is the one about complexity.

SURVEY

The thesis consists of three parts: ‘Theory’, ‘Implementation’ and ‘Applications’.

In the ‘Theory’ part, after having explained the **notation** (chapter 1), we start with **Hilbert-style calculi** (chapter 2) and **possible world semantics** (chapter 3) for the logics K, $\mathsf{K}+T$, KT, $\mathsf{KT}+T$, $\mathsf{S4}$, and K_t. While the Hilbert-style calculi are introduced because they are well-known to many people and they are helpful in one of the applications, possible world semantics will serve as the base for the following chapters.

In order to check whether a formula is valid, we search systematically for a countermodel in the possible world semantics; if we fail, then we know that the formula is valid. We formalise this search with so-called **graph calculi** (chapter 4). The motivation for the graph calculi is exactly the same as for tableau calculi, and the search for a model in a tableau calculus is sometimes presented in similar ways as we present our proofs in graph calculi. The graphical presentation of the calculi themselves, however, is rather unusual. We hope that it helps to clarify the step from possible world semantics to sequent calculi, a step which is in fact well-known, but rarely explained in detail. It will also become clear why there is a simple sequent calculus for the modal logic K, but not for the corresponding tense logic K_t. In this as well as in the following chapter, we first give a detailed presentation for K, and in the following sections, concentrate on the differences between K and the other modal logics.

The standard **sequent calculi** (chapter 5) for K, KT, $\mathsf{S4}$ have been known for a long time, and there exist also tableau calculi for $\mathsf{K}+T$ and $\mathsf{KT}+T$. One problem is that except for K, backward proof search in these calculi does in general not terminate. We develop the sequent calculi further, such that backward proof search always terminates. It is obvious that a loop-check is a solution. The idea to build this loop-check into the sequent calculi seems to be new, but there are some similarities with the algorithms in [Lad77]. Compared to the usual verbal explanation, such a built-in loop-check has two advantages: It is formal and thus more precise, and it makes it easier to discuss optimisations of the loop-check (which do exist). We show how to accelerate the proof search with use-check, which has only been used for classical and intuitionistic logic until now. Note that these are optimisations on a conceptual level, and not on the implementation level. The development of the sequent calculus for K_t is rather complicated. However, in contrast to other sequent calculi that have been proposed, in our calculus the size of a sequent is polynomial with respect to the length of the formula whose validity we want to check, and not exponential. The calculi for K_t are the result of a collaboration with H. Zimmermann.

In the next chapter we investigate the **complexities** (chapter 6) related to backward proof search in the calculi defined so far. Especially between the graph and the sequent calculi, major differences show up, which one does not expect at first sight. We also show that it is not astonishing that the loop check for $\mathsf{K}+T$ cannot be simplified as much as the one for $\mathsf{S4}$.

Finally we investigate **embeddings** (chapter 7) from one logic into another. As in the previous chapter, complexities play an important role.

At the beginning of the 'Implementation' part we give a short introduction to the **Logics Workbench** project (chapter 8). Many people have been involved in the development of the LWB (about fifteen man-years until 1997). The LWB contains implementations of decision procedures for the logics discussed in the 'Theory' part, but it goes much further.

As soon as we had implemented these decision procedures we had to test them. It soon became clear that the formulas one can find in the literature are not sufficient for reliable tests. We hope that the **tests** we present (chapter 9) for K, KT, $\mathsf{S4}$ will help others who are faced with the same problem.

Efficiency is not the only criterion of a decision procedure, but it is certainly an important one. But how should one compare such decision procedures? After discussions with many people we came up with a list of seven postulates for benchmark methods in this area. The **benchmarks** (chapter 10) we propose for K, KT, $\mathsf{S4}$ largely satisfy these postulates.

In the 'Applications' part we show some applications of the decision procedures for K, KT, and $\mathsf{S4}$. First we investigate **relations between propositional normal modal logics** (chapter 11). These relations have been known for a long time, but we present them in a uniform way, such that the results can be checked easily. The search for the formulas required an extensive use of the LWB, but it would not have been possible without the modal logic experts R. Goré and W. Heinle.

In chapter 12 we show how the LWB can be used to compute the **number of formulas** of a fragment of a modal logic. Such computations have only been possible because of the relatively fast decision procedures and since the LWB contains a programming language.

Based on these LWB programs we implement an **optimal simplification** (chapter 13) for formulas in $\mathsf{S4}$. This simplification is of no use in the case of longer formulas, but it is interesting to see that there is for example no formula that is equivalent to $\Box(\Box(p_0 \to \Box p_0) \to p_0) \to p_0$ (in $\mathsf{S4}$) and shorter. We are not aware of any other results of this kind.

Finally we use the LWB to compute some diagrams that correspond to fragments of K, KT, and $\mathsf{S4}$. The one of the positive formulas of $\mathsf{S4}$ looks surprisingly interesting.

RELATED WORK

Of course other calculi and methods for proof search in modal logics have been proposed. We try to give a survey. This survey is certainly incomplete, but the transitive closure of the references will contain most of the relevant work in the area.

In [Fit83] and [Gor] sequent and tableaux calculi for a large number of modal logics are defined. Prefixed tableaux for many logics are discussed in [Mas94] (see [CG97] for an implementation) and [BG97]. In [Hud96] space complexity is the main concern. The influence of the search strategy on the efficiency is investigated in [Dem95] for $\mathsf{S4}$. In the area of description logic, there exist several implementations that can be used as modal logic provers, such as Kris ([BH91]) and FaCT ([Hor97]). See [PC96] and [MMO95] for two further tableaux-like calculi. Tableaux for tense logics are the subject of [RU71] and [Kas94].

Instead of doing backward proof search in sequent calculi it is also possible to do forward proof search or to use a resolution that is based on forward proof search ([Min90], [Vor92]). Another proof search method is the connection method, proposed in [Wal90] for modal logics (see also

[CM97] and [OK96]). In [OS97] embeddings in classical predicate logic are used. See [HS97] for an implementation and a comparison with other methods, including the one in [GS96] that uses a method related to the Davis-Putnam procedure.

A comparison of automated theorem provers for the propositional modal logics K, KT, S4 took place at the conference Tableaux 98. Descriptions of the participating provers, including benchmark results, can be found in [dS98].

TYPESETTING

This thesis was typeset with LaTeX$_{2\epsilon}$, using bibtex and makeindex for the bibliography and the index respectively. Most graphics and pictures have been drawn with the help of LaTeX commands. There are three exceptions which were included as Postscript files: the signature (scanned in), the snapshot of the LWB, and the title pages of the three parts (a Perl script generated three fig files, which were then converted to Postscript). In the third part we include many examples of computations with the LWB. The LaTeX source contains only the input lines. With the help of LaTeX macros (written by Peter Balsiger) and Perl scripts, these input lines are extracted while typesetting, the LWB is started to compute the results, and the output lines are then inserted into the LaTeX file. This method makes sure that the input and output in this thesis really corresponds to the behaviour of the LWB.

THANKS

Everybody who worked on the LWB project, and especially Gerhard Jäger who directed it, has in some way contributed to this thesis. Also the other members of our group, our master thesis students, our guests and many people I met at conferences helped me to understand the problems I discuss. A typical example is the chapter about benchmarks: Only after numerous discussions did I come up with my postulates. I do not try to give a list of all these people, since I would certainly forget some of them; but everybody who had a conversation with me about logic or who is cited in the bibliography can consider himself included. I confine myself to the people who helped me to solve nonlogical problems (in alphabetic order): Peter Balsiger advised me many times concerning LaTeX. Karoline Hunziker-Kanduth tried to improve my English. Gerhard Jäger had the idea for the LWB project and managed to get the fundings. The Swiss National Science Foundation paid me during my work on the LWB and on this thesis (SPP IF 5003-34279 and 21-43197.95).

[illegible]

I

THEORY

1

NOTATION

"Then you should say what you mean," the March Hare went on.
"I do," Alice hastily replied; "at least — at least I mean what I say — that's the same thing, you know."

L. Carroll. Alice's Adventures in Wonderland.

1.1 META-LANGUAGE

1.1.1 DEFINITION meta-logic

In the meta-logic, we use $\Rightarrow$ and $\Leftarrow$ for implication, $\Leftrightarrow$ for equivalence, and $\forall$ and $\exists$ as quantifiers.

1.1.2 DEFINITION sets

In this thesis sets are always enumerable. A set is given either with an enumeration of its elements (for example $\{1, 3, 5 \ldots\}$) or using a condition (for example $\{x \in \mathbb{N} \mid x \text{ is odd}\}$).
Now let S_1, S_2 be two sets.

- $\emptyset$ is the empty set.
- $\text{card}(S)$ is the number of elements in S.
- $x \in S_1$ means that x is an element of S_1.
- $S_1 = S_2$ means that the two sets S_1 and S_2 are equal.
- $S_1 \cup S_2$ is the union of S_1 and S_2.
- $S_1 \subseteq S_2$ means that S_1 is a subset of S_2 (perhaps $S_1 = S_2$).
- $S_1 \times S_2$ is the Cartesian product of S_1 and S_2, i.e. $S_1 \times S_2 = \{\langle x, y\rangle \mid x \in S_1, y \in S_2\}$.

Example

We set $S_1 := \{1,2,3\}$ and $S_2 := \{1,4,5\}$. Then we have: $2 \in S_1$, $2 \notin S_2$, $S_1 \cup S_2 = \{1,2,3,4,5\}$, $\{1,3,2,3\} = S_1$, $\emptyset \subseteq S_1$, $\{1,3\} \subseteq S_1$, $S_1 \subseteq S_1$, $S_1 \times S_2 = \{\langle 1,1\rangle, \langle 1,4\rangle, \langle 1,5\rangle, \langle 2,1\rangle, \langle 2,4\rangle, \langle 2,5\rangle, \langle 3,1\rangle, \langle 3,4\rangle, \langle 3,5\rangle\}$, and $\mathrm{card}(S_2) = 3$.

1.1.3 DEFINITION natural numbers

$\mathbb{N}$ is the set of the natural numbers, i.e. $\mathbb{N} = \{0,1,2,\ldots\}$. We use m and n as meta-variables for natural numbers.

Now let m, n be two natural numbers.

- $m = n$ means that m and n are equal.
- We use $m < n$, $m \leq n$, $m > n$, $m \geq n$ as usual.
- We use $+$ and $\sum$ for sums, and $-$ for subtraction. $\sum_{i\in\emptyset}(n_i) = 0$.
- We use $\cdot$ (sometimes omitted) and $\prod$ for products. m^n stands for $\prod_{i=1}^{n}(m)$.
- $m|n$ means that m divides n.
- $\frac{m}{n}$ is the fraction of m and n.
- $n!$ stands for $1 \cdot 2 \cdot \ldots \cdot n$. $\binom{n}{m}$, where $n \geq m$, stands for $\frac{n!}{m!(n-m)!}$.
- $\max(\{n_1,\ldots,n_m\})$, where $m > 0$, is the maximum of the natural numbers $n_1,\ldots,n_m$.
- $\min(\{n_1,\ldots,n_m\})$, where $m > 0$, is the minimum of the natural numbers $n_1,\ldots,n_m$.
- $\gcd(n_1,n_2)$, where $n_1 \neq 0$ and $n_2 \neq 0$, is the greatest common divisor of the two natural numbers n_1, n_2.

Example

$6|30$, but not $7|30$, $\frac{4}{2} = 2$, $\frac{5}{2} \notin \mathbb{N}$, $\max(\{2,5,3,2\}) = 5$, $\min(\{2,5,3,2\}) = 2$, $\gcd(20,12) = 4$.

1.1.4 DEFINITION functions

We write $f : S_1 \to S_2$ for a function with arguments in the set S_1 and values in the set S_2. Such a function is in general neither surjective nor injective, i.e. in general $\exists y \in S_2 : \forall x \in S_1 : f(x) \neq y$ and $\exists x_1, x_2 \in S_1 : (x_1 \neq x_2$ and $f(x_1) = f(x_2))$.

1.1.5 DEFINITION multisets

We use $\Gamma, \Delta, \Sigma, \Pi, \Lambda$ as meta-variables for multisets of formulas, and H ('history') as a meta-variable for multisets of formulas or multisets of multisets of formulas.

Now let Γ and Δ be two multisets.

- ϵ is the empty multiset.
- $\mathrm{card}(\Gamma)$ is the number of elements in Γ.
- $x \in \Gamma$ means that x is an element of Γ.
- $\Gamma = \Delta$ means that Γ and Δ are equal multisets. This implies that $\mathrm{card}(\Gamma) = \mathrm{card}(\Delta)$.
- Γ, Δ is the multiset union of Γ and Δ.
- x, Γ is the multiset union of the multiset that consists of x and Γ. Γ, x analogous.
- $x_1, \ldots, x_n$ is the multiset that consists of the elements $x_1, \ldots, x_n$.
- $\Gamma \subseteq \Delta$ means Γ is a multiset-subset of Δ. This implies that $\mathrm{card}(\Gamma) \leq \mathrm{card}(\Delta)$.

In general the context will make clear whether a comma stands for multiset union or whether it is just used in an enumeration. Otherwise we write $[\Gamma, \Delta]$, $[x, \Gamma]$, and $[x_1, \ldots, x_n]$ to make clear which comma we mean.

Example

We set $\Gamma := 1, 2, 4$ and $\Delta := 2, 3, 5$. Then $2 \in \Gamma$. Note that $\Gamma = 1, 4, 2$, but $\Gamma \neq 1, 2, 2, 4$. Γ, Δ is the multiset $1, 2, 2, 3, 4, 5$ and $5, \Delta$ is the multiset $2, 3, 5, 5$. A special case is $\Gamma, \epsilon = \Gamma$. We have $1, 2 \subseteq \Gamma$ and $1, 1 \not\subseteq \Gamma$. Finally $\mathrm{card}(\Gamma) = 3$, but $\mathrm{card}(1, 2, 2, 2, 4) = 5$.

1.1.6 DEFINITION tuples

$\langle x_1, \ldots, x_n \rangle$ is the tuple of the elements $x_1, \ldots, x_n$. Let x be a tuple with m and y be a tuple with n elements.

- $x = y$ means that the tuples x and y are equal. This implies that $m = n$ and that $\mathrm{el}_i(x) = \mathrm{el}_i(y)$ for all $1 \leq i \leq n$.
- If x is a tuple with n elements and $1 \leq i \leq n$, then $\mathrm{el}_i(x)$ is the ith element of x.

Example

Let x be the tuple $\langle 2, 1, 3 \rangle$. Then $x \neq \langle 1, 3, 2 \rangle$ and $x \neq \langle 2, 1, 3, 3 \rangle$. We have $\mathrm{el}_1(x) = 2$ and $\mathrm{el}_3(x) = 3$.

1.2 LOGICS IN GENERAL

1.2.1 REMARK logics

L stands for one of the following logics: K, $\mathsf{K} + T$, KT, $\mathsf{KT} + T$, $\mathsf{S4}$, $\mathsf{S4} + T$, K_t, K_n, $\mathsf{K}_n + T$, KT_n, $\mathsf{KT}_n + T$, $\mathsf{S4}_n$, $\mathsf{S4}_n + T$, $\mathsf{S5}$, CPC, IPC, PLTL, and CTL. In some places we also mention linear logic, but with L we never mean linear logic.

1.2.2 REMARK language

The language of a logic L is defined by three sets:

- The set of constants $\{\mathsf{true}, \mathsf{false}\}$.
- A set of unary connectives, including $\neg$.
- A set of binary connectives, including $\wedge, \vee, \rightarrow, \leftrightarrow$.

We use $\bigwedge$ and $\bigvee$ for iterated $\wedge$ and $\vee$, respectively. We always write parentheses around the argument in order to avoid ambiguities. We define $\bigwedge_{i \in \emptyset}(A_i)$ as true and $\bigvee_{i \in \emptyset}(A_i)$ as false.

Example

If $k = 3$, then $(\bigwedge_{i \in \{1, \ldots, k\}}(p_i \rightarrow p_{i+1}) \vee (\neg p_2))$ is $((((p_1 \rightarrow p_2) \wedge (p_2 \rightarrow p_3)) \wedge (p_3 \rightarrow p_4)) \vee (\neg p_2))$ and if $k = 0$, then $(\bigwedge_{i \in \{1, \ldots, k\}}(p_i \rightarrow p_{i+1}) \vee (\neg p_2))$ is $(\mathsf{true} \vee (\neg p_2))$.

1.2.3 DEFINITION variables

For all logics L the set $\mathrm{Var} = \{p_0, p_1, p_2, \ldots\}$ is the set of variables. We use P, Q, R as meta-variables for variables.

1.2.4 DEFINITION formulas

For all logics L we define inductively the set of formulas Fml_L:

- $P \in \mathrm{Var} \;\Rightarrow\; P \in \mathrm{Fml}_L$
- c constant of $L \;\Rightarrow\; c \in \mathrm{Fml}_L$
- $\star$ unary connective of L and $A \in \mathrm{Fml}_L \;\Rightarrow\; (\star A) \in \mathrm{Fml}_L$
- $\circ$ binary connective of L and $A, B \in \mathrm{Fml}_L \;\Rightarrow\; (A \circ B) \in \mathrm{Fml}_L$

We use A, B, C, D, E as meta-variables for formulas, i.e. for elements of Fml_L. The context will make clear which logic L we mean.

In the following we will often make statements like 'A is a $\wedge$ formula'. This means that A is a formula of the form $B \wedge C$. We will use analogous statements for the other unary and binary connectives.

Example

true and $\neg p_0 \to \Box(\Diamond p_2 \vee p_0)$ are formulas of $\mathrm{Fml}_{\mathsf{K}}$.

1.2.5 DEFINITION omitting parentheses in formulas

In order to make formulas more readable we use the following conventions:

- We omit the outermost parentheses.
- $\wedge$, $\vee$, $\leftrightarrow$ are left-associative, and $\to$ is right-associative.
- The unary connectives have a higher priority than the binary ones.
- Sometimes we use also the following order of priorities: $\wedge$ higher than $\vee$ higher than $\to$ higher than $\leftrightarrow$.

Example

In K, $p_0 \vee p_1 \vee p_2$ stands for the formula $((p_0 \vee p_1) \vee p_2)$, $p_0 \to p_1 \to p_2$ stands for the formula $(p_0 \to (p_1 \to p_2))$, and $(\Box p_1 \wedge \neg\Diamond p_2 \to p_0) \vee p_0$ stands for the formula $((((\Box p_1) \wedge (\neg(\Diamond p_2))) \to p_0) \vee p_0)$.

1.2.6 DEFINITION equality of formulas

$A \equiv B$ means that the formulas A and B are syntactically equal.

Example

$\Box p_0 \wedge p_1, p_1 \wedge \Box p_0$ are formulas of K, and $\Box p_0 \wedge p_1 \not\equiv p_1 \wedge \Box p_0$.

1.2.7 DEFINITION substitution, instances of formulas

If $P_0, \ldots, P_n$ are different variables and A is a formula with $\mathrm{vars}(A) \subseteq \{P_0, \ldots, P_n\}$, then $A\{C_0/P_0, \ldots, C_n/P_n\}$ is the formula A with C_0 substituted for P_0, …, C_n substituted for P_n (all substitutions in parallel). We then call $A\{C_0/P_0, \ldots, C_n/P_n\}$ an instance of the formula A.

Example

If $A \equiv \Box(p_0 \to p_1 \wedge p_2) \vee \neg p_0$, then $A\{p_3 \vee p_1/p_0\} \equiv \Box(p_3 \vee p_1 \to p_1 \wedge p_2) \vee \neg(p_3 \vee p_1)$ and $A\{p_3 \vee p_1/p_0, p_4/p_1\} \equiv \Box(p_3 \vee p_1 \to p_4 \wedge p_2) \vee \neg(p_3 \vee p_1)$.

1.2.8 DEFINITION formula schemes

Formula schemes are defined in the same way as formulas, but with the letters $z_0, z_1, \ldots$ instead of $p_0, p_1, \ldots$. To obtain the formula scheme that corresponds to a formula, we replace p_i by z_i for all $i \in \mathbb{N}$. The notation with $z_0, z_1, \ldots$ is tiresome and unusual. Therefore we will write the letters A, B, C, D (sometimes with subscripts) instead of $z_0, z_1, \dots$. If it is not clear from the context that we mean a formula scheme and not a formula, then we will explicitly say so.

Example

$p_0 \wedge \Box(p_0 \to p_1) \to p_1$ is a formula, and $z_0 \wedge \Box(z_0 \to z_1) \to z_1$ is the corresponding formula scheme. In the following we will write $A \wedge \Box(A \to B) \to B$ instead.

1.2.9 DEFINITION schemes of multisets of formulas

A formula scheme stands for a certain class of formulas. Analogously, a scheme of a multiset of formulas stands for a certain class of multisets. Sometimes we also use schemes of multisets of multisets of formulas.

This definition is rather informal, since we think that a formalisation would be more confusing than helpful.

Example

$A \wedge B, \Gamma$ is a scheme of a multiset of formulas of K. The multiset $\Box p_0 \wedge p_1, \Diamond(\mathsf{true} \vee p_1), p_3 \to \neg p_0$ is an instance of this scheme. For this instance we have $\Gamma = \Diamond(\mathsf{true} \vee p_1), p_3 \to \neg p_0$.

$\Box A, \Diamond\Delta, \Gamma$ is another scheme of a multiset of formulas of K. An instance of this scheme is for example the multiset $\Box\neg p_1, \Diamond p_0, \Diamond(p_1 \leftrightarrow \Box p_2), \Box p_3, \mathsf{false}$. Another instance is the multiset $\Box\mathsf{true}$, where both Δ and Γ are empty.

1.2.10 DEFINITION theories

A theory is a finite multiset of formulas. Th_L is the set of all theories of the logic L.

We use T as a meta-variable for theories.

1.2.11 DEFINITION vars

For each logic L we define inductively the function $\mathrm{vars}(A) : \mathrm{Fml}_L \to \mathrm{Var}$:

- $P \in \mathrm{Var} \;\Rightarrow\; \mathrm{vars}(P) := \{P\}$
- c constant of $L \;\Rightarrow\; \mathrm{vars}(c) := \emptyset$
- $\star$ unary connective of L, $A \in \mathrm{Fml}_L \;\Rightarrow\; \mathrm{vars}(\star A) := \mathrm{vars}(A)$
- $\circ$ binary connective of L and $A, B \in \mathrm{Fml}_L \;\Rightarrow\; \mathrm{vars}(A \circ B) := \mathrm{vars}(A) \cup \mathrm{vars}(B)$

$\mathrm{vars}(A)$ is the set of variables that occur in the formula A.

Example

In K we have $\mathrm{vars}(\Box(\mathsf{true} \to p_3 \wedge (p_1 \leftrightarrow p_3))) = \{p_1, p_3\}$.

1.2.12 DEFINITION subfmls

For each logic L we define inductively the function $\mathrm{subfmls}(A) : \mathrm{Fml}_L \to \mathrm{Th}_L$:

- $P \in \mathrm{Var} \;\Rightarrow\; \mathrm{subfmls}(P) := \{P\}$
- c constant of $L \;\Rightarrow\; \mathrm{subfmls}(c) := \{c\}$
- $\star$ unary connective of L, $A \in \mathrm{Fml}_L \;\Rightarrow\; \mathrm{subfmls}(\star A) := \{\star A\} \cup \mathrm{subfmls}(A)$
- $\circ$ binary connective of L and $A, B \in \mathrm{Fml}_L$
  $\Rightarrow\; \mathrm{subfmls}(A \circ B) := \{A \circ B\} \cup \mathrm{subfmls}(A) \cup \mathrm{subfmls}(B)$

subfmls(A) is the set of the subformulas of the formula A.

Example

In K we have $\text{subfmls}(\Box(\mathsf{true} \rightarrow p_3 \wedge p_3)) = \{\mathsf{true}, p_3, p_3 \wedge p_3, \mathsf{true} \rightarrow p_3 \wedge p_3, \Box(\mathsf{true} \rightarrow p_3 \wedge p_3)\}$.

1.2.13 DEFINITION length

For each logic L we define inductively the function $\text{length}(A) : \text{Fml}_L \rightarrow \mathbb{N}$:

- $P \in \text{Var} \;\Rightarrow\; \text{length}(P) := 1$
- c constant of $L \;\Rightarrow\; \text{length}(c) := 1$
- $\star$ unary connective of L, $A \in \text{Fml}_L \;\Rightarrow\; \text{length}(\star A) := 1 + \text{length}(A)$
- $\circ$ binary connective of L and $A, B \in \text{Fml}_L$
  $\Rightarrow\; \text{length}(A \circ B) := 1 + \text{length}(A) + \text{length}(B)$

length(A) is the length of the formula A.
If T is the multiset of formulas $A_1, \ldots, A_n$, then $\text{length}(T) := \sum_{i=1}^{n}(\text{length}(A_i))$.

Example

In K we have $\text{length}(\Box(\mathsf{true} \rightarrow p_3 \wedge p_3)) = 6$, and $\text{length}(\Box p_1, p_2 \rightarrow \neg p_0, \mathsf{true}) = 7$.

1.2.14 DEFINITION depth

For each logic L we define inductively the function $\text{depth}(A) : \text{Fml}_L \rightarrow \mathbb{N}$:

- $P \in \text{Var} \;\Rightarrow\; \text{depth}(P) := 0$
- c constant of $L \;\Rightarrow\; \text{depth}(c) := 0$
- $\star$ unary connective of L, $A \in \text{Fml}_L \;\Rightarrow\; \text{depth}(\star A) := 1 + \text{depth}(A)$
- $\circ$ binary connective of L and $A, B \in \text{Fml}_L$
  $\Rightarrow\; \text{depth}(A \circ B) := 1 + \max(\{\text{depth}(A), \text{depth}(B)\})$

depth(A) is the depth of the formula A.

Example

In K we have $\text{depth}(\Box(\mathsf{true} \rightarrow p_3 \wedge p_3)) = 3$.

1.2.15 DEFINITION modaldepth

For each logic $L \in \{\mathsf{K}, \mathsf{K} + T, \mathsf{KT}, \mathsf{KT} + T, \mathsf{S4}, \mathsf{S4} + T, \mathsf{K}_t\}$ we define inductively the function $\text{modaldepth}(A) : \text{Fml}_L \rightarrow \mathbb{N}$:

- $P \in \text{Var} \;\Rightarrow\; \text{modaldepth}(P) := 0$
- c constant of $L \;\Rightarrow\; \text{modaldepth}(c) := 0$
- $\star$ unary connective of L, $\star \neq \neg$, and $A \in \text{Fml}_L$
  $\Rightarrow\; \text{modaldepth}(\star A) := 1 + \text{modaldepth}(A)$
- $A \in \text{Fml}_L \;\Rightarrow\; \text{modaldepth}(\neg A) := \text{modaldepth}(A)$
- $\circ$ binary connective of L and $A, B \in \text{Fml}_L$
  $\Rightarrow\; \text{modaldepth}(A \circ B) := \max(\{\text{modaldepth}(A), \text{modaldepth}(B)\})$

modaldepth(A) is the modal depth of the formula A.

Example

In K we have modaldepth$(\Box(\Diamond\mathsf{true} \to p_3 \wedge \Box p_3)) = 2$.

1.3 K, $\mathsf{K}+T$, KT, $\mathsf{KT}+T$, $\mathsf{S4}$, $\mathsf{S4}+T$

1.3.1 DEFINITION language

set of constants: $\{\mathsf{true}, \mathsf{false}\}$

set of unary connectives: $\{\Box, \Diamond, \neg\}$

set of binary connectives: $\{\wedge, \vee, \to, \leftrightarrow\}$

1.3.2 DEFINITION $\Box\Gamma$, $\Diamond\Gamma$

If $\Gamma = A_1, \dots, A_n$, then $\Box\Gamma$ is the multiset $\Box A_1, \dots, \Box A_n$ and $\Diamond\Gamma$ is the multiset $\Diamond A_1, \dots, \Diamond A_n$. Especially $\Box\epsilon$ is the multiset ϵ and $\Diamond\epsilon$ is the multiset ϵ.

1.3.3 DEFINITION $\Box^n A$, $\Diamond^n A$

We use the following notation for iterated $\Box$ and $\Diamond$:

- $\Box^0 A :\equiv A, \quad \Box^{n+1} A :\equiv \Box\Box^n A$
- $\Diamond^0 A :\equiv A, \quad \Diamond^{n+1} A :\equiv \Diamond\Diamond^n A$

1.3.4 DEFINITION nnf

For each logic $L \in \{\mathsf{K}, \mathsf{K}+T, \mathsf{KT}, \mathsf{KT}+T, \mathsf{S4}, \mathsf{S4}+T\}$ we define inductively the function $\mathrm{nnf}(A) : \mathrm{Fml}_L \to \mathrm{Fml}_L$:

- $P \in \mathrm{Var} \;\Rightarrow\; \mathrm{nnf}(P) :\equiv P$, $\mathrm{nnf}(\neg P) :\equiv \neg P$
- $\mathrm{nnf}(\mathsf{true}) :\equiv \mathsf{true}$, $\mathrm{nnf}(\mathsf{false}) :\equiv \mathsf{true}$, $\mathrm{nnf}(\neg\mathsf{true}) :\equiv \mathsf{false}$, $\mathrm{nnf}(\neg\mathsf{false}) :\equiv \mathsf{true}$
- $A \in \mathrm{Fml}_L \;\Rightarrow\; \mathrm{nnf}(\Box A) :\equiv \Box\mathrm{nnf}(A)$, $\mathrm{nnf}(\Diamond A) :\equiv \Diamond\mathrm{nnf}(A)$, $\mathrm{nnf}(\neg\Box A) :\equiv \Diamond\mathrm{nnf}(\neg A)$, $\mathrm{nnf}(\neg\Diamond A) :\equiv \Box\mathrm{nnf}(\neg A)$
- $A \in \mathrm{Fml}_L \;\Rightarrow\; \mathrm{nnf}(\neg\neg A) :\equiv \mathrm{nnf}(A)$
- $A, B \in \mathrm{Fml}_L \;\Rightarrow\; \mathrm{nnf}(A \wedge B) :\equiv \mathrm{nnf}(A) \wedge \mathrm{nnf}(B)$, $\mathrm{nnf}(A \vee B) :\equiv \mathrm{nnf}(A) \vee \mathrm{nnf}(B)$, $\mathrm{nnf}(\neg(A \wedge B)) :\equiv \mathrm{nnf}(\neg A \vee \neg B)$, $\mathrm{nnf}(\neg(A \vee B)) :\equiv \mathrm{nnf}(\neg A \wedge \neg B)$
- $A, B \in \mathrm{Fml}_L \;\Rightarrow\; \mathrm{nnf}(A \to B) :\equiv \mathrm{nnf}(\neg A \vee B)$, $\mathrm{nnf}(\neg(A \to B)) :\equiv \mathrm{nnf}(A \wedge \neg B)$
- $A, B \in \mathrm{Fml}_L \;\Rightarrow\; \mathrm{nnf}(A \leftrightarrow B) :\equiv \mathrm{nnf}((A \to B) \wedge (B \to A))$, $\mathrm{nnf}(\neg(A \leftrightarrow B)) :\equiv \mathrm{nnf}(\neg((A \to B) \wedge (B \to A)))$

If T is the multiset $B_1, \dots, B_n$, then $\mathrm{nnf}(\neg T)$ is an abbreviation for the multiset $\mathrm{nnf}(\neg B_1)$, ..., $\mathrm{nnf}(\neg B_n)$.

A formula is in negation normal form if it contains neither $\to$ nor $\leftrightarrow$ and if $\neg$ occurs only in front of variables. Note that $\mathrm{nnf}(A)$ is in negation normal form.

Example

In K, $\mathrm{nnf}(p_0 \to \Box\Diamond\neg(p_1 \wedge \neg\neg\Box p_2)) \equiv \neg p_0 \vee \Box\Diamond(\neg p_1 \vee \Diamond\neg p_2)$ and $\mathrm{nnf}(\neg\neg p_0 \to \mathsf{false}, \neg\Box p_0)$ is the multiset $\neg p_0 \vee \mathsf{false}, \Diamond\neg p_0$.

1.3.5 DEFINITION standard formulas

In figure 1.a we introduce abbreviations for formulas in $\mathrm{Fml}_{\mathsf{K}}$. The literature contains many variants of the formulas. See chapter 11 for more information.

1.4 K_t

1.4.1 DEFINITION language

set of constants: $\{\mathsf{true}, \mathsf{false}\}$

set of unary connectives: $\{\Box, \Diamond, \blacksquare, \blacklozenge, \neg\}$

set of binary connectives: $\{\wedge, \vee, \to, \leftrightarrow\}$

1.4.2 DEFINITION $\Box\Gamma$, $\blacksquare\Gamma$, $\Diamond\Gamma$, $\blacklozenge\Gamma$

If $\Gamma = A_1, \dots, A_n$, then $\Box\Gamma$ is the multiset $\Box A_1, \dots, \Box A_n$ and $\blacksquare\Gamma$ is the multiset $\blacksquare A_1, \dots, \blacksquare A_n$. Analogously $\Diamond\Gamma$ is the multiset $\Diamond A_1, \dots, \Diamond A_n$ and $\blacklozenge\Gamma$ is the multiset $\blacklozenge A_1, \dots, \blacklozenge A_n$. Especially $\Box\epsilon = \blacksquare\epsilon = \Diamond\epsilon = \blacklozenge\epsilon = \epsilon$.

1.4.3 DEFINITION nnf

We define inductively the function $\mathrm{nnf}(A) : \mathrm{Fml}_{\mathsf{K}_t} \to \mathrm{Fml}_{\mathsf{K}_t}$:

- $P \in \mathrm{Var} \;\Rightarrow\; \mathrm{nnf}(P) :\equiv P$, $\mathrm{nnf}(\neg P) :\equiv \neg P$
- $\mathrm{nnf}(\mathsf{true}) :\equiv \mathsf{true}$, $\mathrm{nnf}(\mathsf{false}) :\equiv \mathsf{false}$, $\mathrm{nnf}(\neg\mathsf{true}) :\equiv \mathsf{false}$,
- $\mathrm{nnf}(\neg\mathsf{false}) :\equiv \mathsf{true}$
- $A \in \mathrm{Fml}_L \;\Rightarrow\; \mathrm{nnf}(\Box A) :\equiv \Box\mathrm{nnf}(A)$, $\mathrm{nnf}(\Diamond A) :\equiv \Diamond\mathrm{nnf}(A)$, $\mathrm{nnf}(\blacksquare A) :\equiv \blacksquare\mathrm{nnf}(A)$, $\mathrm{nnf}(\blacklozenge A) :\equiv \blacklozenge\mathrm{nnf}(A)$ $\mathrm{nnf}(\neg\Box A) :\equiv \Diamond\mathrm{nnf}(\neg A)$, $\mathrm{nnf}(\neg\Diamond A) :\equiv \Box\mathrm{nnf}(\neg A)$, $\mathrm{nnf}(\neg\blacksquare A) :\equiv \blacklozenge\mathrm{nnf}(\neg A)$, $\mathrm{nnf}(\neg\blacklozenge A) :\equiv \blacksquare\mathrm{nnf}(\neg A)$
- $A \in \mathrm{Fml}_L \;\Rightarrow\; \mathrm{nnf}(\neg\neg A) :\equiv \mathrm{nnf}(A)$
- $A, B \in \mathrm{Fml}_L \;\Rightarrow\; \mathrm{nnf}(A \wedge B) :\equiv \mathrm{nnf}(A) \wedge \mathrm{nnf}(B)$, $\mathrm{nnf}(A \vee B) :\equiv \mathrm{nnf}(A) \vee \mathrm{nnf}(B)$, $\mathrm{nnf}(\neg(A \wedge B)) :\equiv \mathrm{nnf}(\neg A \vee \neg B)$, $\mathrm{nnf}(\neg(A \vee B)) :\equiv \mathrm{nnf}(\neg A \wedge \neg B)$
- $A, B \in \mathrm{Fml}_L \;\Rightarrow\; \mathrm{nnf}(A \to B) :\equiv \mathrm{nnf}(\neg A \vee B)$, $\mathrm{nnf}(\neg(A \to B)) :\equiv \mathrm{nnf}(A \wedge \neg B)$
- $A, B \in \mathrm{Fml}_L \;\Rightarrow\; \mathrm{nnf}(A \leftrightarrow B) :\equiv \mathrm{nnf}((A \to B) \wedge (B \to A))$, $\mathrm{nnf}(\neg(A \leftrightarrow B)) :\equiv \mathrm{nnf}(\neg((A \to B) \wedge (B \to A)))$

A formula is in negation normal form if it contains neither $\to$ nor $\leftrightarrow$ and if $\neg$ occurs only in front of variables. The formula $\mathrm{nnf}(A)$ is in negation normal form. We will call $\mathrm{nnf}(A)$ the negation normal form of the formula A.

name	formula
D	$\Box p_0 \to \Diamond p_0$
D_2	$\Diamond\mathsf{true}$
T	$\Box p_0 \to p_0$
4	$\Box p_0 \to \Box\Box p_0$
4_{M}	$\Box p_0 \wedge \Diamond p_1 \to \Diamond(\Box p_0 \wedge p_1)$
5	$\Diamond p_0 \to \Box\Diamond p_0$
5_{M}	$\Diamond p_0 \wedge \Diamond p_1 \to \Diamond(\Diamond p_0 \wedge p_1)$
B	$p_0 \to \Box\Diamond p_0$
B_{M}	$p_0 \wedge \Diamond p_1 \to \Diamond(\Diamond p_0 \wedge p_1)$
G	$\Diamond\Box p_0 \to \Box\Diamond p_0$
G_0	$\Diamond(p_0 \wedge \Box p_1) \to \Box(p_0 \vee \Diamond p_1)$
H	$\Box(p_0 \vee p_1) \wedge \Box(\Box p_0 \vee p_1) \wedge \Box(p_0 \vee \Box p_1) \to \Box p_0 \vee \Box p_1$
H^+	$\Box(\Box p_0 \vee p_1) \wedge \Box(p_0 \vee \Box p_1) \to \Box p_0 \vee \Box p_1$
L	$\Box(p_0 \wedge \Box p_0 \to p_1) \vee \Box(p_1 \wedge \Box p_1 \to p_0)$
L^+	$\Box(\Box p_0 \to p_1) \vee \Box(\Box p_1 \to p_0)$
L^{++}	$\Box(\Box p_0 \to \Box p_1) \vee \Box(\Box p_1 \to \Box p_0)$
M	$\Box\Diamond p_0 \to \Diamond\Box p_0$
M_2	$\Diamond\Box(p_0 \to \Box p_0)$
M_3	$\Box\Diamond p_0 \wedge \Box\Diamond p_1 \to \Diamond(p_0 \wedge p_1)$
Pt	$\Box(p_0 \vee \Diamond p_0) \to \Diamond(p_0 \wedge \Box p_0)$
W	$\Box(\Box p_0 \to p_0) \to \Box p_0$
W_0	$\Box\Diamond\mathsf{true} \to \Box\mathsf{false}$
Z	$\Box(\Box p_0 \to p_0) \to (\Diamond\Box p_0 \to \Box p_0)$
Dum	$\Box(\Box(p_0 \to \Box p_0) \to p_0) \to (\Diamond\Box p_0 \to p_0)$
Dum_1	$\Box(\Box(p_0 \to \Box p_0) \to p_0) \to (\Diamond\Box p_0 \to \Box p_0)$
Dum_2	$\Box(\Box(p_0 \to \Box p_0) \to \Box p_0) \to (\Diamond\Box p_0 \to p_0)$
Dum_3	$\Box(\Box(p_0 \to \Box p_0) \to \Box p_0) \to (\Diamond\Box p_0 \to \Box p_0)$
Dum_4	$\Box(\Box(p_0 \to \Box p_0) \to p_0) \to (\Diamond\Box p_0 \to p_0 \vee \Box p_0)$
Grz	$\Box(\Box(p_0 \to \Box p_0) \to p_0) \to p_0$
Grz_1	$\Box(\Box(p_0 \to \Box p_0) \to p_0) \to \Box p_0$
Grz_2	$\Box(\Box(p_0 \to \Box p_0) \to \Box p_0) \to p_0$
Grz_3	$\Box(\Box(p_0 \to \Box p_0) \to \Box p_0) \to \Box p_0$
Grz_4	$\Box(\Box(p_0 \to \Box p_0) \to p_0) \to p_0 \vee \Box p_0$
Grz_5	$\Box(\Box(p_0 \to \Box p_1) \to \Box p_1) \wedge \Box(\Box(\neg p_0 \to \Box p_1) \to \Box p_1) \to \Box p_1$
F	$(\Diamond\Box p_0 \to p_1) \vee \Box(\Box p_1 \to p_0)$
H_{s}	$p_0 \to \Box(\Diamond p_0 \to p_0)$
P	$\Diamond\Box\Diamond p_0 \to (p_0 \to \Box p_0)$
R	$\Diamond\Box p_0 \to (p_0 \to \Box p_0)$
X	$\Box\Box p_0 \to \Box p_0$
Zem	$\Box\Diamond\Box p_0 \to (p_0 \to \Box p_0)$

Figure 1.a: Standard formulas.

1.5 OTHER LOGICS

1.5.1 DEFINITION language of $\mathsf{S5}$

Same language as for K.

1.5.2 DEFINITION language of K_n, $\mathsf{K}_n + T$, KT_n, $\mathsf{KT}_n + T$, $\mathsf{S4}_n$, $\mathsf{S4}_n + T$

Same language as K, but with $\Box$ and $\Diamond$ replaced by the (infinitely many) connectives $\Box_0$, $\Diamond_0$, $\Box_1$, $\Diamond_1$, ...

1.5.3 DEFINITION language of CPC

Language of K without $\Box$ and $\Diamond$.

1.5.4 DEFINITION language of IPC

Language of K without $\Box$ and $\Diamond$.

1.5.5 DEFINITION language of PLTL

constants: $\{\mathsf{true}, \mathsf{false}\}$
unary connectives: $\{\neg, \Box, \Diamond, \mathsf{X}\}$
binary connectives: $\{\wedge, \vee, \rightarrow, \leftrightarrow, \mathsf{B}, \mathsf{U}\}$
U is the strong until, and B is the weak before.

1.5.6 DEFINITION language of CTL

constants: $\{\mathsf{true}, \mathsf{false}\}$
unary connectives: $\{\neg, \mathsf{AF}, \mathsf{AG}, \mathsf{AX}, \mathsf{EF}, \mathsf{EG}, \mathsf{EX}\}$
binary connectives: $\{\wedge, \vee, \rightarrow, \leftrightarrow, \mathsf{AU}, \mathsf{AB}, \mathsf{EU}, \mathsf{EB}\}$

2

HILBERT-STYLE CALCULI

Hil·bert (hílbərt), David (1862–1943). German mathematician whose principal researches were on the theory of numbers, foundations of geometry and integral equations. He belonged to the formalist school of scientific logic

Webster Encyclopedic Dictionary.

2.1 INTRODUCTION

2.1.1 REMARK why Hilbert-style calculi?

In this chapter we define the Hilbert-style calculi for the logics we investigate in the following. Since Hilbert-style calculi are fairly well-known, we hope this helps the reader to understand which logics we denote with the names K, $\mathsf{K}+T$, KT, $\mathsf{KT}+T$, $\mathsf{S4}$, $\mathsf{S4}+T$, K_t.

We do not search for proofs in these calculi. Because of the missing subformula property (see remark 2.1.6), we do not believe that such an attempt could be more successful — for the logics we investigate — than the proof search based on sequent calculi. Things look different if no 'reasonable' sequent calculus is known for a logic (see [Mor76]). (Strictly speaking, not in every calculus without subformula property is backward proof search inefficient, see for example [SS92].)

2.1.2 DEFINITION Hilbert-style calculus

A Hilbert-style calculus for the logic L consists of:

- An enumerable set of axioms.

  An axiom has the form $\frac{}{A}$, where A is a formula scheme of L.
- An enumerable set of rules, including modus ponens.

  A rule has the form $\frac{A}{C}$ or $\frac{A \quad B}{C}$, where A, B, C are formula schemes of L. Modus ponens is the rule $\frac{A \quad A \to B}{B}$.

- A finite (perhaps empty) multiset of formulas of L (called 'additional assumptions' in the following).

Example

$\frac{}{\Box A \to A}$, where $\Box A \to A$ is a formula scheme, is an axiom of the Hilbert-style calculus $\mathsf{KT}^{\mathcal{H}}$. $\frac{}{\Box p_0 \to p_0}$, $\frac{}{\Box\neg\neg p_1 \to \neg\neg p_1}$, $\frac{}{\Box(p_3 \land \Diamond p_2) \to (p_3 \land \Diamond p_2)}$ are instances of this axiom.

$\frac{A \quad A\to B}{B}$ is a rule of the Hilbert-style calculus $\mathsf{KT}^{\mathcal{H}}$. $\frac{p_3 \quad p_3 \to p_0}{p_0}$ and $\frac{p_3 \quad p_3 \to \Diamond(p_3 \land p_2)}{\Diamond(p_3 \land p_2)}$ are instances of this rule.

The calculus $\mathsf{KT}^{\mathcal{H}}$ defined in definition 2.4.2 is a Hilbert-style calculus without additional assumptions. The calculus $(\mathsf{KT} + T)^{\mathcal{H}}$ defined in definition 2.5.1 is the corresponding Hilbert-style calculus with additional assumptions.

2.1.3 DEFINITION $L^{\mathcal{H}} \vdash C$

Let $L^{\mathcal{H}}$ be a Hilbert-style calculus for L. We define inductively $L^{\mathcal{H}} \vdash C$ for $C \in \mathrm{Fml}_L$:

- If $\frac{}{C}$ is an instance of an axiom of $L^{\mathcal{H}}$, then $L^{\mathcal{H}} \vdash C$.
- If C is one of the additional assumptions of $L^{\mathcal{H}}$, then $L^{\mathcal{H}} \vdash C$.
- If $\frac{A}{C}$ is an instance of a rule of $L^{\mathcal{H}}$ and $L^{\mathcal{H}} \vdash A$, then $L^{\mathcal{H}} \vdash C$.
- There $\frac{A \quad B}{C}$ is an instance of a rule of $L^{\mathcal{H}}$ and $L^{\mathcal{H}} \vdash A$, $L^{\mathcal{H}} \vdash B$, then $L^{\mathcal{H}} \vdash C$.

If $L^{\mathcal{H}} \vdash C$, then we say that C is provable in $L^{\mathcal{H}}$.

Example

See the example in definition 2.2.1 for a proof in a Hilbert-style calculus without additional assumptions, and the example in definition 2.3.1 for a proof in a Hilbert-style calculus with additional assumptions.

2.1.4 REMARK additional assumptions vs. axioms

Note that an additional assumption is just a formula, whereas the set of all instances of an axiom is in general infinite. Therefore it can be sensible to add p_0 as an additional assumption (see for example the example in 2.3.1). However, if we add the formula scheme A as an axiom, then all formulas become provable, since every formula is an instance of the axiom A.

2.1.5 THEOREM forward proof search

Let $L^{\mathcal{H}}$ be a Hilbert-style calculus for the logic L. We can enumerate all formulas that are provable in $L^{\mathcal{H}}$. In this way we obtain a semi-decision procedure. This procedure returns the value 'provable' if its input is a provable formula, and it does not terminate otherwise.

Proof

There exists an enumeration of the formulas of L. Since the axioms are enumerable and there are only finitely many additional assumptions, we can also enumerate the formulas that are provable in one step in $L^{\mathcal{H}}$ (say C_0, C_1, ...). Since the rules are enumerable, too, we can enumerate the formulas that have proofs in $L^{\mathcal{H}}$ where the leaves of the proof tree are in $\{C_0, C_1, \ldots, C_n\}$ and the proof tree has at most depth n.

2.1.6 REMARK backward proof search

Assume that we have to decide whether a given formula $A \in \mathrm{Fml}_L$ is provable in a Hilbert-style calculus $L^{\mathcal{H}}$ for L.

If $L^{\mathcal{H}} \vdash A$, then there must exist a proof for A, i.e. A is either an instance of an axiom (decidable in the Hilbert-style calculi we are considering), or it was obtained by the application of a rule.

If the last step was an application of (mp), then the set of possible pairs of premises is infinite, namely $\langle B, B \to A\rangle$ for all $B \in \mathrm{Fml}_L$. This problem occurs because the rule (mp) does not satisfy the subformula property: The formula B occurs in the premises but not in the conclusion.

Thus we have an infinite branching degree at every step when doing backward proof search. A depth-first search is hopeless. With breadth-first search, we obtain a semi-decision procedure (if a formula is provable, then the search stops, otherwise it does not stop), but not a decision procedure.

2.1.7 REMARK rule (□) vs. axiom $\frac{}{A \to \Box A}$

Note that the rule (□) introduced in the next section and the axiom $\frac{}{A \to \Box A}$ are not 'equivalent'. For $L \in \{\mathsf{K}, \mathsf{KT}, \mathsf{S4}, \mathsf{K}_t\}$ we have $L^{\mathcal{H}} \nvdash p_0 \to \Box p_0$, although (□) is a rule of $L^{\mathcal{H}}$. If we replaced for example in $\mathsf{KT}^{\mathcal{H}}$ the rule (□) by the axiom $\frac{}{A \to \Box A}$, then we could even show $\mathsf{KT}^{\mathcal{H}} \vdash A \leftrightarrow \Box A$, i.e. the connective □ would become meaningless.

2.2 K

2.2.1 DEFINITION Hilbert-style calculus $\mathsf{K}^{\mathcal{H}}$

axioms:

If $A \in \mathrm{Fml}_{\mathsf{CPC}}$ A is valid in classical propositional logic, and B is the formula scheme of K that corresponds to A, then $\frac{}{B}$(cpc) is an axiom.

$$\frac{}{\Box A \leftrightarrow \neg\Diamond\neg A}\ (\Diamond\Box_1) \qquad \frac{}{\Box\neg A \leftrightarrow \neg\Diamond A}\ (\Diamond\Box_2)$$

$$\frac{}{\Box(A \to B) \to (\Box A \to \Box B)}\ (\mathrm{k})$$

rules:

$$\frac{A \quad A \to B}{B}\ (\mathrm{mp}) \qquad \frac{A}{\Box A}\ (\Box)$$

Example

$\mathsf{K}^{\mathcal{H}} \vdash \Box p_0 \to \Box(p_0 \vee \Diamond p_1)$, as the following proof shows.

$$\dfrac{\dfrac{\dfrac{}{p_0 \to p_0 \vee \Diamond p_1}\ (\mathrm{cpc})}{\Box(p_0 \to p_0 \vee \Diamond p_1)}\ (\Box) \qquad \Box(p_0 \to p_0 \vee \Diamond p_1) \to (\Box p_0 \to \Box(p_0 \vee \Diamond p_1))}{\Box p_0 \to \Box(p_0 \vee \Diamond p_1)}\ (\mathrm{mp})$$

The formula $p_6 \to (p_6 \vee p_7)$ is an element of $\mathrm{Fml}_{\mathsf{CPC}}$ and valid in classical propositional logic. Therefore the formula scheme $A \to (A \vee B)$ is an axiom of $\mathsf{K}^{\mathcal{H}}$. The formula $p_0 \to (p_0 \vee \Diamond p_1)$ is an instance of this axiom, and thus $\mathsf{K}^{\mathcal{H}} \vdash p_0 \to (p_0 \vee \Diamond p_1)$.

$$\dfrac{\dfrac{\dfrac{\vdots}{A} \quad \dfrac{\vdots}{A \to B}}{B}\,(\text{mp})}{\Box B}\,(\Box) \quad \rightsquigarrow \quad \dfrac{\dfrac{\dfrac{\vdots}{A}}{\Box A}\,(\Box) \quad \dfrac{\dfrac{\dfrac{\vdots}{A \to B}}{\Box(A \to B)}\,(\Box) \quad \dfrac{}{\Box(A \to B) \to (\Box A \to \Box B)}\,(\text{k})}{\Box A \to \Box B}\,(\text{mp})}{\Box B}\,(\text{mp})$$

Figure 2.a: The transformations used in the proof of theorem 2.3.2. to move $(\Box)$ applications upwards.

2.3 $\mathsf{K} + T$

2.3.1 DEFINITION Hilbert-style calculus $(\mathsf{K} + T)^{\mathcal{H}}$

$(\mathsf{K} + T)^{\mathcal{H}}$ is the calculus $\mathsf{K}^{\mathcal{H}}$ plus the elements of T as additional assumptions.

Example

$(\mathsf{K} + [p_0, p_0 \to \Box\neg p_1])^{\mathcal{H}} \vdash \Box\neg\Diamond p_1$, as the following proof shows.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{p_0} \quad \dfrac{}{p_0 \to \Box\neg p_1}}{\Box\neg p_1}\,(\text{mp}) \quad \dfrac{\dfrac{}{\Box\neg p_1 \leftrightarrow \neg\Diamond p_1}\,(\Diamond\Box_2) \quad \dfrac{}{(\Box\neg p_1 \leftrightarrow \neg\Diamond p_1) \to (\Box\neg p_1 \to \neg\Diamond p_1)}\,(\text{cpc})}{\Box\neg p_1 \to \neg\Diamond p_1}\,(\text{mp})}{\neg\Diamond p_1}\,(\text{mp})}{\Box\neg\Diamond p_1}\,(\Box)$$

Note that $(\mathsf{K} + [p_0])^{\mathcal{H}} \vdash p_0$, but $(\mathsf{K} + [p_0])^{\mathcal{H}} \nvdash p_2$, i.e. substitution is not an admissible rule.

2.3.2 THEOREM normal form of proofs in $(\mathsf{K} + T)^{\mathcal{H}}$

If $(\mathsf{K}+T)^{\mathcal{H}} \vdash A$ for some theory T, then there is a proof $\mathcal{P}$ of A in $(\mathsf{K}+T)^{\mathcal{H}}$ such that $(\Box)$ is only applied on formulas of the form $\Box^m C$, where $\dfrac{}{C}$ is an instance of an axiom or C is an additional assumption of $(\mathsf{K} + T)^{\mathcal{H}}$.

Proof

With the transformation in figure 2.a we can move the application of $(\Box)$ upwards. Finally we obtain a proof of the desired form.

2.4 KT

2.4.1 REMARK from $\mathsf{K}^{\mathcal{H}}$ to $\mathsf{KT}^{\mathcal{H}}$

Compared with $\mathsf{K}^{\mathcal{H}}$, the calculus $\mathsf{KT}^{\mathcal{H}}$ contains an additional axiom, namely $\dfrac{}{\Box A \to A}$.

In terms of the possible world semantics, the axiom says: If A holds in all successor worlds, then it also holds in the current world, i.e. the current world belongs to the successor worlds.

Consequently for each world there exists at least one successor world, namely the world itself. Thus the axiom corresponds to the reflexivity of the accessibility relation in the possible world semantics for KT (cp. section 3.4).

2.4.2 DEFINITION Hilbert-style calculus $\mathsf{KT}^{\mathcal{H}}$

axioms:

If $A \in \mathrm{Fml}_{\mathsf{CPC}}$ A is valid in classical propositional logic, and B is the formula scheme of KT that corresponds to A, then $\dfrac{}{B}\,(\text{cpc})$ is an axiom.

$$\dfrac{}{\Box A \leftrightarrow \neg\Diamond\neg A}\,(\Diamond\Box_1) \qquad \dfrac{}{\Box\neg A \leftrightarrow \neg\Diamond A}\,(\Diamond\Box_2)$$

$$\dfrac{}{\Box(A \to B) \to (\Box A \to \Box B)}\,(\text{k}) \qquad \dfrac{}{\Box A \to A}\,(\text{t})$$

rules:

$$\dfrac{A \quad A \to B}{B}\,(\text{mp}) \qquad \dfrac{A}{\Box A}\,(\Box)$$

Example

$\mathsf{KT}^{\mathcal{H}} \vdash \Box\Box(p_1 \land p_0) \to \Box(p_0 \land p_1)$, as the following proof shows.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\begin{array}{l}\Box(p_1\land p_0)\\ \to p_1\land p_0\end{array}}\,(\text{t}) \quad \dfrac{}{\begin{array}{l}(\Box(p_1 \land p_0) \to p_1 \land p_0)\\ \to (\Box(p_1\land p_0) \to p_0\land p_1)\end{array}}\,(\text{cpc})}{\Box(p_1 \land p_0) \to p_0 \land p_1}\,(\text{mp})}{\Box(\Box(p_1\land p_0) \to p_0\land p_1)}\,(\Box) \quad \dfrac{}{\begin{array}{l}\Box(\Box(p_1 \land p_0) \to p_0 \land p_1)\\ \to (\Box\Box(p_1 \land p_0) \to \Box(p_0 \land p_1))\end{array}}\,(\text{k})}{\Box\Box(p_1 \land p_0) \to \Box(p_0 \land p_1)}\,(\text{mp})$$

2.5 $\mathsf{KT} + T$

2.5.1 DEFINITION Hilbert-style calculus $(\mathsf{KT} + T)^{\mathcal{H}}$

$(\mathsf{KT} + T)^{\mathcal{H}}$ is the calculus $\mathsf{KT}^{\mathcal{H}}$ plus the elements of T as additional assumptions.

Example

$(\mathsf{KT} + [p_0, \Box p_1])^{\mathcal{H}} \vdash \Box(p_0 \land p_1)$, as the following proof shows.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Box p_1} \quad \dfrac{}{\Box p_1 \to p_1}\,(\text{t})}{p_1}\,(\text{mp}) \quad \dfrac{\dfrac{}{p_0} \quad \dfrac{}{p_0 \to (p_1 \to (p_0 \land p_1))}\,(\text{cpc})}{p_1 \to (p_0 \land p_1)}\,(\text{mp})}{p_0 \land p_1}\,(\text{mp})}{\Box(p_0 \land p_1)}\,(\Box)$$

2.6 S4

2.6.1 REMARK from $\mathsf{KT}^{\mathcal{H}}$ to $\mathsf{S4}^{\mathcal{H}}$

Compared with $\mathsf{KT}^{\mathcal{H}}$, the calculus $\mathsf{S4}^{\mathcal{H}}$ contains an additional axiom, namely $\frac{}{\Box A \to \Box\Box A}$. Instead of $\frac{}{\Box A \to \Box\Box A}$ we could also add $\frac{}{\Diamond\Diamond A \to \Diamond A}$.
In terms of the possible world semantics, the axiom says: If a world is accessible in two steps, then it is accessible in one step. Thus the axiom corresponds to the transitivity of the accessibility relation in the possible world semantics for $\mathsf{S4}$ (cp. section 3.6).

2.6.2 DEFINITION Hilbert-style calculus $\mathsf{S4}^{\mathcal{H}}$

axioms:

If $A \in \mathrm{Fml}_{\mathsf{CPC}}$ A is valid in classical propositional logic, and B is the formula scheme that corresponds to A, then $\frac{}{B}$(cpc) is an axiom.

$$\frac{}{\Box A \leftrightarrow \neg\Diamond\neg A}\ (\Diamond\Box_1) \qquad \frac{}{\Box\neg A \leftrightarrow \neg\Diamond A}\ (\Diamond\Box_2)$$

$$\frac{}{\Box(A \to B) \to (\Box A \to \Box B)}\ (\mathrm{k}) \qquad \frac{}{\Box A \to A}\ (\mathrm{t})$$

$$\frac{}{\Box A \to \Box\Box A}\ (4)$$

rules:

$$\frac{A \quad A \to B}{B}\ (\mathrm{mp}) \qquad \frac{A}{\Box A}\ (\Box)$$

Example

$\mathsf{S4}^{\mathcal{H}} \vdash \Box(\neg\Box\Box\neg p_1 \to \neg\Box\neg p_1)$, as the following proof shows.

$$\cfrac{\cfrac{\cfrac{}{\Box\neg p_1 \to \Box\Box\neg p_1}\,(4) \qquad \cfrac{}{(\Box\neg p_1 \to \Box\Box\neg p_1) \to (\neg\Box\Box\neg p_1 \to \neg\Box\neg p_1)}\,(\mathrm{cpc})}{\neg\Box\Box\neg p_1 \to \neg\Box\neg p_1}\,(\mathrm{mp})}{\Box(\neg\Box\Box\neg p_1 \to \neg\Box\neg p_1)}\,(\Box)$$

2.7 $\mathsf{S4} + T$

2.7.1 DEFINITION Hilbert-style calculus $(\mathsf{S4} + T)^{\mathcal{H}}$

$(\mathsf{S4} + T)^{\mathcal{H}}$ is the calculus $\mathsf{S4}^{\mathcal{H}}$ plus the elements of T as additional assumptions.

2.7.2 THEOREM embedding of $(\mathsf{S4} + T)^{\mathcal{H}}$ in $\mathsf{S4}^{\mathcal{H}}$

$$(\mathsf{S4} + [B_1, \dots, B_n])^{\mathcal{H}} \vdash A \quad \Leftrightarrow \quad \mathsf{S4}^{\mathcal{H}} \vdash \Box B_1 \to \dots \to \Box B_n \to A$$

Proof

We show $(\mathsf{S4}+T')^{\mathcal{H}} \vdash A \Leftrightarrow (\mathsf{S4}+T)^{\mathcal{H}} \vdash \Box B_{m+1} \to A$, where $T = [B_1, \dots, B_m]$ and $T' = [B_1, \dots, B_m, B_{m+1}]$. Then the theorem follows with an induction on the number of elements in the theory and applications of (cpc) and (mp). Note that we do not add instances of the axiom (t) in our transformation.

'$\Leftarrow$':

Obviously $(\mathsf{S4}+T')^{\mathcal{H}} \vdash B_{m+1}$ and thus $(\mathsf{S4}+T')^{\mathcal{H}} \vdash \Box B_{m+1}$. Since $T \subseteq T'$ we know $(\mathsf{S4}+T')^{\mathcal{H}} \vdash \Box B_{m+1} \to A$. With an application of (mp) follows $(\mathsf{S4}+T')^{\mathcal{H}} \vdash A$.

'$\Rightarrow$':

Let $\mathcal{P}$ be a proof of A in $(\mathsf{S4}+T')^{\mathcal{H}}$. We replace each formula C in $\mathcal{P}$ by $\Box B_{m+1} \to C$. The resulting $\mathcal{P}'$ is not a proof. However, using the transformations from figure 2.b we can convert $\mathcal{P}'$ into a proof of $\Box B_{m+1} \to A$ in $(\mathsf{S4}+T)^{\mathcal{H}}$.

2.8 K_t

2.8.1 REMARK from $\mathsf{K}^{\mathcal{H}}$ to $\mathsf{K}_t^{\mathcal{H}}$

$\mathsf{K}_t^{\mathcal{H}}$ contains two copies of $\mathsf{K}^{\mathcal{H}}$, one with $\Box$, $\Diamond$ and one with $\blacksquare$, $\blacklozenge$. The additional axioms $\blacklozenge\Box A \to A$ and $\Diamond\blacksquare A \to A$ link the connectives $\Box$, $\Diamond$ with the connectives $\blacksquare$, $\blacklozenge$. See section 3.8 for their meaning in possible world semantics.

2.8.2 DEFINITION Hilbert-style calculus $\mathsf{K}_t^{\mathcal{H}}$

axioms:

If $A \in \mathrm{Fml}_{\mathsf{CPC}}$ and $\mathsf{CPC} \vdash A$, and B is the formula scheme of K_t that corresponds to A, then $\frac{}{B}$(cpc) is an axiom.

$$\frac{}{\Box A \leftrightarrow \neg\Diamond\neg A}\ (\Diamond\Box_1) \qquad \frac{}{\Box\neg A \leftrightarrow \neg\Diamond A}\ (\Diamond\Box_2)$$

$$\frac{}{\blacksquare A \leftrightarrow \neg\blacklozenge\neg A}\ (\blacklozenge\blacksquare_1) \qquad \frac{}{\blacksquare\neg A \leftrightarrow \neg\blacklozenge A}\ (\blacklozenge\blacksquare_2)$$

$$\frac{}{\Box(A \to B) \to (\Box A \to \Box B)}\ (\mathrm{k}) \qquad \frac{}{\blacksquare(A \to B) \to (\blacksquare A \to \blacksquare B)}\ (\mathrm{k}\blacksquare)$$

$$\frac{}{\blacklozenge\Box A \to A}\ (\blacklozenge\Box) \qquad \frac{}{\Diamond\blacksquare A \to A}\ (\Diamond\blacksquare)$$

rules:

$$\frac{A \quad A \to B}{B}\ (\mathrm{mp})$$

$$\frac{A}{\Box A}\ (\Box) \qquad \frac{A}{\blacksquare A}\ (\blacksquare)$$

$$\dfrac{\dfrac{\vdots}{\Box B_{m+1}\to C}\quad \dfrac{\vdots}{\Box B_{m+1}\to (C\to D)}}{\Box B_{m+1}\to D} \quad\rightsquigarrow\quad \dfrac{\dfrac{\vdots}{\Box B_{m+1}\to C}\quad \dfrac{\dfrac{\vdots}{\Box B_{m+1}\to (C\to D)}\quad \dfrac{}{\begin{array}{l}(\Box B_{m+1}\to (C\to D))\\ \to ((\Box B_{m+1}\to C)\to (\Box B_{m+1}\to D))\end{array}}\,(\mathrm{cpc})}{(\Box B_{m+1}\to C)\to (\Box B_{m+1}\to D)}\,(\mathrm{mp})}{\Box B_{m+1}\to D}\,(\mathrm{mp})$$

$$\dfrac{\dfrac{\vdots}{\Box B_{m+1}\to C}}{\Box B_{m+1}\to \Box C} \quad\rightsquigarrow\quad \dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\Box B_{m+1}\to C}}{\Box(\Box B_{m+1}\to C)}\,(\Box)\quad \dfrac{}{\begin{array}{l}\Box(\Box B_{m+1}\to C)\\ \to (\Box\Box B_{m+1}\to \Box C)\end{array}}\,(\mathrm{k})}{\Box\Box B_{m+1}\to \Box C}\,(\mathrm{mp})\quad \dfrac{\dfrac{}{\begin{array}{l}\Box B_{m+1}\\ \to \Box\Box B_{m+1}\end{array}}\,(4)\quad \dfrac{}{\begin{array}{l}(\Box B_{m+1}\to \Box\Box B_{m+1})\\ \to ((\Box\Box B_{m+1}\to \Box C)\\ \to (\Box B_{m+1}\to \Box C))\end{array}}\,(\mathrm{cpc})}{(\Box\Box B_{m+1}\to \Box C)\to (\Box B_{m+1}\to \Box C)}\,(\mathrm{mp})}{\Box B_{m+1}\to \Box C}\,(\mathrm{mp})$$

Figure 2.b: The transformations used in the proof of theorem 2.7.2.

2.9 OTHER LOGICS

2.9.1 REMARK other modal logics

See chapter 11 for Hilbert-style calculi for many other propositional normal modal logics.

2.9.2 REMARK $\mathsf{S5}$

We obtain a Hilbert-style calculus for $\mathsf{S5}$ by adding the axiom $\frac{}{\neg\Box A \to \Box\neg\Box A}$ to $\mathsf{S4}^{\mathcal{H}}$.

2.9.3 REMARK K_n, $\mathsf{K}_n + T$, KT_n, $\mathsf{KT}_n + T$, $\mathsf{S4}_n$, $\mathsf{S4}_n + T$

See [FHMV96] or [HM92] for Hilbert-style calculi for these multimodal logics.

2.9.4 REMARK other multimodal logics

In [FHMV96] and [HM92] Hilbert-style calculi for many other multimodal logics are defined.

2.9.5 REMARK CPC, IPC

See for example [TvD88] for a Hilbert-style calculus for IPC. If we add the axiom $A \vee \neg A$ to this calculus, then we obtain a Hilbert-style calculus for CPC.

2.9.6 REMARK PLTL

In [Eme90] a Hilbert-style calculus for PLTL is defined.

2.9.7 REMARK linear logic

In [Tro92] a Hilbert-style calculus for linear logic is defined.

2.10 SUMMARY

We have defined the usual Hilbert-style calculi for the logics K, $\mathsf{K} + T$, KT, $\mathsf{KT} + T$, $\mathsf{S4}$, $\mathsf{S4} + T$, K_t. In addition, we have proved constructively a normal form theorem for proofs in $(\mathsf{K} + T)^{\mathcal{H}}$ and an embedding of $\mathsf{S4} + T$ in $\mathsf{S4}$.

3

POSSIBLE WORLD SEMANTICS

When George is hanged Harris will be the worst packer in this world; and I looked at the piles of plates and cups, and kettles, and bottles, and jars, and pies, and stoves, and cakes, and tomatoes, etc., and felt that the thing would soon become exciting.
It did. They started by breaking a cup. That was the first thing they did. They did that just to show you what they *could* do, and to get you interested.

J.K. Jerome. Three Men in a Boat.

3.1 INTRODUCTION

3.1.1 REMARK from CPC to modal logics

In the usual semantics of classical propositional logic, a valuation assigns 0 or 1 to each variable. For a given valuation we can compute the value (0 or 1) of a formula of $\mathrm{Fml}_{\mathsf{CPC}}$.

In the possible world semantics, we have not just one valuation, but several valuations. Each valuation corresponds to a so-called world. These worlds are linked by a so-called accessibility relation. The value of $\Box A$ is 1 in a world w if the value of A is 1 in each world that is accessible from w (i.e. in all successor worlds). The value of $\Diamond A$ is 1 in a world w if the value of A is 1 in at least one world that is accessible from w. In the case of K_t, we also have connectives $\blacksquare$ and $\blacklozenge$ that say the same for the predecessor worlds.

The only difference between the possible world semantics for K, KT, $\mathsf{S4}$ are the conditions that are imposed on the accessibility relation.

3.1.2 REMARK additional assumptions

Later on we will define when a formula is valid in K, KT, and $\mathsf{S4}$. In the definition of the validity of a formula in $\mathsf{K}+T$, $\mathsf{KT}+T$, $\mathsf{S4}+T$, we take the possible world semantics of the corresponding logic without theory and demand in addition that for each formula $A \in T$, the value of the formula A must be 1 in each world.

3.1.3 REMARK number of models

Note that in the case of CPC, there is only a finite number of different 'relevant' valuations, whereas in the case of possible world semantics we have an infinite number of models, even if we restrict ourselves to a finite number of valuations. As a further restriction we can try to limit the number of worlds in the models; this is possible, but much more difficult.

3.1.4 DEFINITION satisfiable

A formula A is satisfiable in a logic L if $\neg A$ is not valid in L.

3.1.5 CONVENTION meta-variable for models

We use $\mathcal{M}$ as a meta-variable for models.

3.2 K

3.2.1 DEFINITION K frame

A K frame is a pair $\langle W, \mathcal{R} \rangle$, where:

- W is a non-empty set.
- $\mathcal{R} \subseteq W \times W$.

3.2.2 DEFINITION K model

A K model is a triple $\langle W, \mathcal{R}, v \rangle$, where:

- $\langle W, \mathcal{R} \rangle$ is a K frame.
- $v : W \times \mathrm{Var} \to \{0, 1\}$.

We call the elements of W the worlds, $\mathcal{R}$ the accessibility relation, and v the valuation of the K model.

Example

Assume that:

- $W = \{w_0, w_1, w_2\}$.
- $w_0 \mathcal{R} w_1$, $w_0 \mathcal{R} w_2$, $w_1 \mathcal{R} w_1$.
- $v(w_0, p_0) = 1$, $v(w_0, p_2) = 1$, $v(w_1, p_0) = 1$, 0 otherwise.

Then $\langle W, \mathcal{R}, v \rangle$ is a K model.

3.2.3 REMARK represent K models

In order to represent finite K models we use circles for the worlds and arrows for the accessibility relation. In the circles we list those variables that are true in this world. Sometimes we will also write negated variables inside circles in order to stress that they are false.

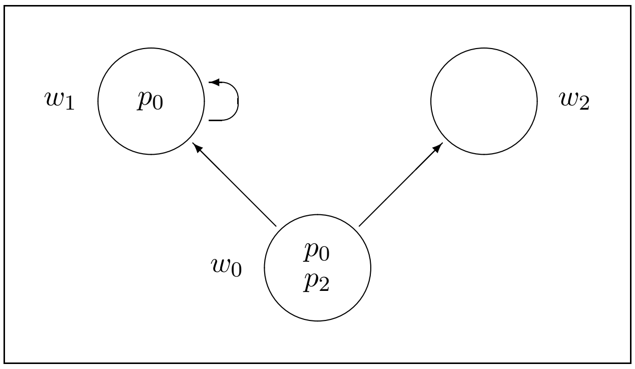


Figure 3.a: The K model of the example in definition 3.2.2.

Example

See figure 3.a for the representation of the K model defined in the example in definition 3.2.2.

3.2.4 DEFINITION diameter of a K model

Assume that $\mathcal{M} = \langle W, \mathcal{R}, v\rangle$ is a K model. The tuple $\langle w_0, \ldots, w_n\rangle \in W^n$ is a path in $\mathcal{M}$ iff:

- $w_0 \mathcal{R} w_1$, $w_1 \mathcal{R} w_2$, $\ldots$, $w_{n-1} \mathcal{R} w_n$
- $\forall i, j \in \{0, \ldots, n\} : (i \neq j \Rightarrow w_i \neq w_j)$

For all $w, w' \in W$ such that there exists a path from w to w' in $\mathcal{M}$ we define $|w, w'|$, the length of the shortest path from w to w'.

$$|w, w'| := \min\{n + 1 \mid \langle w_0, \ldots, w_n\rangle \text{ path in } \mathcal{M}, w_0 = w, w_n = w'\}$$

Now we can define $\operatorname{diam}(\mathcal{M})$, the diameter of $\mathcal{M}$. If $\mathcal{R} = \emptyset$, then $\operatorname{diam}(\mathcal{M}) := 0$. Otherwise we set:

$$\operatorname{diam}(\mathcal{M}) := \max\{|w, w'| \mid w, w' \in W\}$$

Example

Let $\mathcal{M}$ be the K model represented in figure 3.b. Then the following tuples are the paths in $\mathcal{M}$:

- $\langle w_0\rangle$, $\langle w_0, w_1\rangle$, $\langle w_0, w_1, w_2\rangle$, $\langle w_0, w_2\rangle$, $\langle w_0, w_3\rangle$
- $\langle w_1\rangle$, $\langle w_1, w_2\rangle$
- $\langle w_2\rangle$
- $\langle w_3\rangle$, $\langle w_3, w_0\rangle$, $\langle w_3, w_0, w_1\rangle$, $\langle w_3, w_0, w_1, w_2\rangle$, $\langle w_3, w_0, w_2\rangle$

Note that for example $\langle w_0, w_3, w_0\rangle$ is not a path. Now we compute the diameter of $\mathcal{M}$. We have:

- $|w_0, w_0| = 1$, $|w_0, w_1| = 2$, $|w_0, w_2| = 2$, $|w_0, w_3| = 2$
- $|w_1, w_1| = 1$, $|w_1, w_2| = 2$
- $|w_2, w_2| = 1$
- $|w_3, w_3| = 1$, $|w_3, w_0| = 2$, $|w_3, w_1| = 3$, $|w_3, w_2| = 3$

Thus $\operatorname{diam}(\mathcal{M}) = \max(\{1, 2, 2, 2, 1, 2, 1, 1, 2, 3, 3\}) = 3$.

3.2.5 DEFINITION $\mathsf{K}, \mathcal{M} \models A$

Let $\langle W, \mathcal{R}, v\rangle$ be a K model. We extend the valuation v to a function $W \times \mathrm{Fml}_{\mathsf{K}} \to \{0, 1\}$.

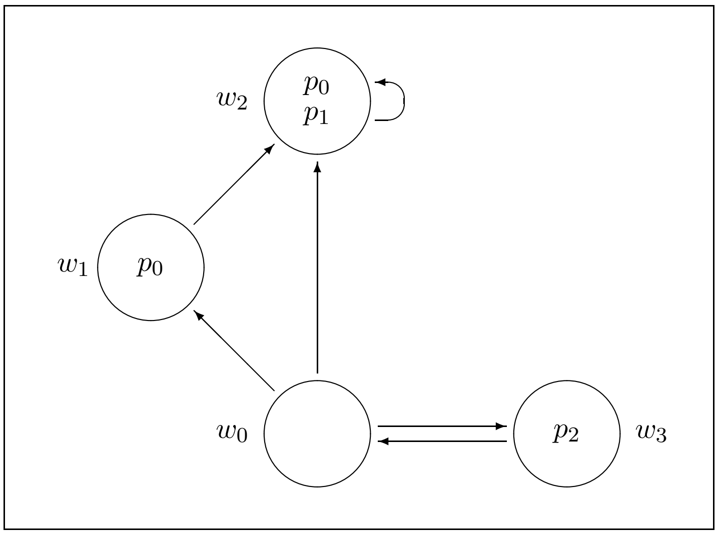


Figure 3.b: A K model with diameter 3.

1. $v(w, \mathsf{true}) := 1$
2. $v(w, \mathsf{false}) := 0$
3. $v(w, \Box A) := \begin{cases} 1 & \forall w' \in W : (w\mathcal{R}w' \Rightarrow v(w', A) = 1) \\ 0 & \text{otherwise} \end{cases}$
4. $v(w, \Diamond A) := \begin{cases} 1 & \exists w' \in W : (w\mathcal{R}w' \text{ and } v(w', A) = 1) \\ 0 & \text{otherwise} \end{cases}$
5. $v(w, \neg A) := 1 - v(w, A)$
6. $v(w, A \wedge B) := \min(v(w, A), v(w, B))$
7. $v(w, A \vee B) := \max(v(w, A), v(w, B))$
8. $v(w, A \rightarrow B) := v(w, \neg A \vee B)$
9. $v(w, A \leftrightarrow B) := \begin{cases} 1 & v(w, A) = v(w, B) \\ 0 & \text{otherwise} \end{cases}$

If $\langle W, \mathcal{R}, v\rangle$ is a K model, then:

$$\mathsf{K}, \langle W, \mathcal{R}, v\rangle \models A \quad :\Leftrightarrow \quad \forall w \in W : v(w, A) = 1$$

Example

In the K model $\mathcal{M}$ represented in figure 3.a we have:

- $v(w_0, p_0) = 1,\ v(w_0, p_2) = 1,\ v(w_0, p_0 \wedge p_2) = 1$
- $v(w_1, p_0) = 1,\ v(w_1, p_2) = 0,\ v(w_1, p_0 \wedge p_2) = 0$
- $v(w_0, \Box p_0) = 0,\ v(w_0, \Box\Box p_0) = 1$
- $v(w_0, \Diamond p_0) = 1,\ v(w_0, \Diamond\Diamond p_0) = 1,\ v(w_0, \Diamond p_2) = 0$

Moreover, $\mathsf{K}, \mathcal{M} \not\models p_0 \vee \Diamond p_0$, but $\mathsf{K}, \mathcal{M} \models p_2 \vee \Diamond p_0$.

3.2.6 DEFINITION $\mathsf{K} \models A$

$$\mathsf{K} \models A \quad :\Leftrightarrow \quad \text{for all } \mathsf{K} \text{ models } \mathcal{M} : \mathsf{K}, \mathcal{M} \models A$$

If $\mathsf{K} \models A$, then we say that A is valid in K.

Example

Since the accessibility relation in K models is in general not reflexive, we have $\mathsf{K} \not\models \Diamond(p_0 \vee \neg p_0)$. Proof: If $\mathcal{M} = \langle \{w_0\}, \emptyset, v\rangle$, then $\mathsf{K}, \mathcal{M} \not\models \Diamond(p_0 \vee \neg p_0)$ for all valuations v.

However, $\mathsf{K} \models \Diamond(p_0 \vee \neg p_0) \vee \Box p_1$. Proof: Let w be a world in an arbitrary K model $\langle W, \mathcal{R}, v\rangle$. If there is a world w' with $w\mathcal{R}w'$, then $v(w', p_0 \vee \neg p_0) = 1$, thus $v(w, \Diamond(p_0 \vee \neg p_0)) = 1$. If there is no such world, then $v(w, \Box p_1) = 1$.

3.3 $\mathsf{K}+T$

3.3.1 DEFINITION $\mathsf{K}+T$ model

$\mathcal{M}$ is a $\mathsf{K}+T$ model iff:

- $\mathcal{M}$ is a K model
- $\forall B \in T : \mathsf{K}, \mathcal{M} \models B$

3.3.2 DEFINITION $\mathsf{K}+T \models A$

$$\mathsf{K}+T \models A \quad :\Leftrightarrow \quad \text{for all } \mathsf{K}+T \text{ models } \mathcal{M} : \mathsf{K}, \mathcal{M} \models A$$

If $\mathsf{K}+T \models A$, then we say that A is valid in $\mathsf{K}+T$.

3.4 KT

3.4.1 DEFINITION KT frame

A KT frame is a pair $\langle W, \mathcal{R}\rangle$, where:

- W is a non-empty set.
- $\mathcal{R} \subseteq W \times W$ and $\mathcal{R}$ is reflexive (i.e. $\forall w \in W : w\mathcal{R}w$).

3.4.2 DEFINITION KT model

We just replace 'K frame' by 'KT frame' in the definition 3.2.2 of K models.

Example

Assume that:

- $W = \{w_0, w_1, w_2\}$
- $w_0\mathcal{R}w_1$, $w_0\mathcal{R}w_2$, $w_2\mathcal{R}w_0$, $\forall w \in W : w\mathcal{R}w$
- $v(w_0, p_0) = 1$, $v(w_1, p_0) = 1$, $v(w_1, p_2) = 1$, 0 otherwise

Then $\langle W, \mathcal{R}, v\rangle$ is a KT model. $\langle W, \mathcal{R}', v\rangle$, where $w_0\mathcal{R}'w_1$, $w_0\mathcal{R}'w_2$, $w_0\mathcal{R}'w_0$, $w_2\mathcal{R}'w_2$, is not a KT model, since the relation $\mathcal{R}'$ is not reflexive.

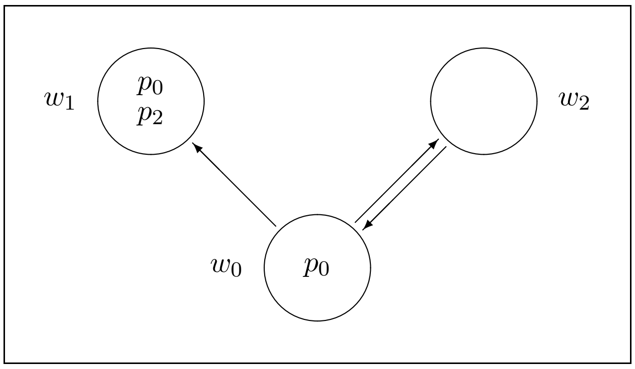


Figure 3.c: The KT model of the example in definition 3.4.2.

3.4.3 REMARK represent KT models

We represent finite KT models as we represent finite K models, but we omit the arrows from a world to itself.

Example

See figure 3.c for the representation of the KT model of the example in definition 3.4.2.

3.4.4 DEFINITION $\mathsf{KT} \models A$

We just replace 'K' by 'KT' in the definitions of $\mathsf{K}, \mathcal{M} \models A$ and of $\mathsf{K} \models A$. If $\mathsf{KT} \models A$, then we say that A is valid in KT.

Example

For the KT model defined in the example in definition 3.4.2 we have:

- $v(w_0, \Box p_0) = 0,\ v(w_0, \Box\Box p_0) = 0$
- $v(w_1, \Diamond p_2) = 1$
- $v(w_2, \Diamond p_2) = 0$
- $v(w_0, \Diamond(\Diamond p_2 \vee p_4)) = 1$

and $\mathsf{KT}, \langle W, \mathcal{R}, v\rangle \models \Diamond\Diamond p_2$.

In contrast to K models, the accessibility relation in KT models is reflexive. Therefore we have $\mathsf{KT} \models \Diamond(p_0 \vee \neg p_0)$.

Since the accessibility relation is not transitive, $\mathsf{KT} \not\models \Box\neg p_2 \rightarrow \Box\Box\neg p_2$. Proof: In the KT model of the example in definition 3.4.2 (see also figure 3.c) we have $v(w_2, \Box\neg p_2) = 1$, but $v(w_2, \Box\Box\neg p_2) = 0$.

3.5 $\mathsf{KT} + T$

3.5.1 DEFINITION $\mathsf{KT} + T$ model

$\mathcal{M}$ is a $\mathsf{KT} + T$ model iff:

- $\mathcal{M}$ is a KT model.
- $\forall B \in T : \mathsf{KT}, \mathcal{M} \models B$.

3.5.2 DEFINITION $\mathsf{KT}+T \models A$

$$\mathsf{KT}+T \models A \quad :\Leftrightarrow \quad \text{for all } \mathsf{KT}+T \text{ models } \mathcal{M} : \mathsf{KT}, \mathcal{M} \models A$$

If $\mathsf{KT}+T \models A$, then we say that A is valid in $\mathsf{KT}+T$.

3.6 S4

3.6.1 DEFINITION S4 frame

An S4 frame is a pair $\langle W, \mathcal{R}\rangle$, where:

- W is a non-empty set.
- $\mathcal{R} \subseteq W \times W$, and $\mathcal{R}$ is reflexive (i.e. $\forall w \in W : w\mathcal{R}w$) and transitive (i.e. $\forall w_1, w_2, w_3 \in W : (w_1\mathcal{R}w_2$ and $w_2\mathcal{R}w_3 \Rightarrow w_1\mathcal{R}w_3))$.

3.6.2 DEFINITION S4 model

We just replace 'K frame' by 'S4 frame' in the definition 3.2.2 of K models.

Example

Assume that:

- $W = \{w_0, w_1, w_2\}$
- $w_0\mathcal{R}w_1$, $w_0\mathcal{R}w_2$, $w_1\mathcal{R}w_2$, $\forall w \in W : w\mathcal{R}w$
- $v(w_0, p_0) = 1$, $v(w_1, p_0) = 1$, $v(w_1, p_3) = 1$, 0 otherwise

Then $\langle W, \mathcal{R}, v\rangle$ is a S4 model. $\langle W, \mathcal{R}', v\rangle$, where $w_0\mathcal{R}'w_1$, $w_1\mathcal{R}'w_2$, $\forall w \in W : w\mathcal{R}'w$, is not a S4 model, since the relation $\mathcal{R}'$ is not transitive.

3.6.3 REMARK represent S4 models

We represent finite S4 models as we represent finite K models, but we omit the arrows that follow from the other arrows because of the reflexivity and transitivity of the accessibility relation.

Example

See figure 3.d for the representation of the S4 model of the example in definition 3.6.2.

3.6.4 DEFINITION $\mathsf{S4} \models A$

We just replace 'K' by 'S4' in the definitions of $\mathsf{K}, \mathcal{M} \models A$ and of $\mathsf{K} \models A$. If $\mathsf{S4} \models A$, then we say that A is valid in S4.

Example

For the S4 model represented in figure 3.d we have:

- $v(w_0, \Diamond p_0) = 1,\ v(w_0, \Diamond\neg p_0) = 1$
- $v(w_2, \Diamond p_0) = 0,\ v(w_1, \Diamond\neg p_0) = 1$

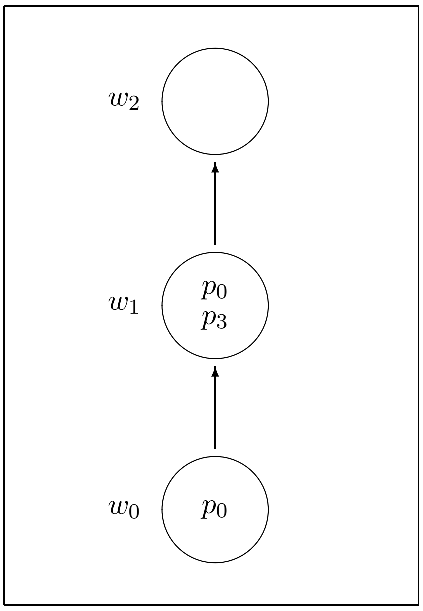


Figure 3.d: The $\mathsf{S4}$ model of the example in definition 3.6.2.

Since the accessibility relation is transitive, we have $\mathsf{S4} \models \Box p_1 \to \Box\Box p_1$. Proof: Let w be a world in a $\mathsf{S4}$ model $\langle W, \mathcal{R}, v\rangle$. If $v(w, \Box p_1) = 1$, then also $v(w, \Box\Box p_1) = 1$, since every world that is accessible from w is accessible in one step.

3.7 $\mathsf{S4} + T$

3.7.1 DEFINITION $\mathsf{S4} + T$ model

$\mathcal{M}$ is a $\mathsf{S4} + T$ model iff:

- $\mathcal{M}$ is a $\mathsf{S4}$ model.
- $\forall B \in T : \mathsf{S4}, \mathcal{M} \models B$

3.7.2 DEFINITION $\mathsf{S4} + T \models A$

$$\mathsf{S4} + T \models A \quad :\Leftrightarrow \quad \text{for all } \mathsf{S4} + T \text{ models } \mathcal{M} : \mathsf{S4}, \mathcal{M} \models A$$

If $\mathsf{S4} + T \models A$, then we say that A is valid in $\mathsf{S4} + T$.

3.7.3 THEOREM embedding of $\mathsf{S4} + T$ in $\mathsf{S4}$

$$\mathsf{S4} + [B_1, \ldots, B_n] \models A \quad \Leftrightarrow \quad \mathsf{S4} \models \Box B_1 \wedge \ldots \wedge \Box B_n \to A$$

Proof

'$\Leftarrow$':

Let $\mathcal{M} = \langle W, \mathcal{R}, v\rangle$ be an $\mathsf{S4} + [B_1, \ldots, B_n]$ model. Then $\forall i \in \{1, \ldots, n\} : \forall w \in W : v(w, \Box B_i) = 1$. Since $\mathsf{S4} \models \Box B_1 \wedge \ldots \wedge \Box B_n \to A$ we also know that $\forall w \in W : v(\Box B_1 \wedge \ldots \wedge \Box B_n \to A) = 1$.

Thus $\forall w \in W : v(w, A) = 1$, i.e. $\mathcal{M} \models A$. Since $\mathcal{M}$ is an arbitrary $\mathsf{S4} + [B_1, \ldots, B_n]$ model it follows $\mathsf{S4} + [B_1, \ldots, B_n] \models A$.

'$\Rightarrow$':

Let $\mathcal{M} = \langle W, \mathcal{R}, v\rangle$ be an $\mathsf{S4}$ model and $w \in W$. If $v(w, \Box B_1 \wedge \ldots \wedge \Box B_n) = 0$, then $v(w, \Box B_1 \wedge \ldots \wedge \Box B_n \to A) = 1$. Otherwise $\mathcal{M}$ restricted to the worlds accessible from w is a $\mathsf{S4}+[B_1, \ldots, B_n]$ model and therefore $v(w, A) = 1$. Since $\mathcal{M}$ is an arbitrary $\mathsf{S4}$ model it follows $\mathsf{S4} \models \Box B_1 \wedge \ldots \wedge \Box B_n \to A$.

3.8 K_t

3.8.1 DEFINITION K_t frame, K_t model

A K_t frame is a K frame, and K_t model is a K model.

3.8.2 REMARK represent K_t models

We represent finite K_t models in the same way as finite K models.

3.8.3 DEFINITION $\mathsf{K}_t, \mathcal{M} \models A$

Let $\langle W, \mathcal{R}, v\rangle$ is a K_t model. We extend the valuation v to a function $W \times \mathrm{Fml}_{\mathsf{K}_t} \to \{0, 1\}$.

1. $v(w, \mathsf{true}) := 1$
2. $v(w, \mathsf{false}) := 0$
3. $v(w, \Box A) := \begin{cases} 1 & \forall w' \in W : (w\mathcal{R}w' \Rightarrow v(w', A) = 1) \\ 0 & \text{otherwise} \end{cases}$
4. $v(w, \Diamond A) := \begin{cases} 1 & \exists w' \in W : (w\mathcal{R}w' \text{ and } v(w', A) = 1) \\ 0 & \text{otherwise} \end{cases}$
5. $v(w, \blacksquare A) := \begin{cases} 1 & \forall w' \in W : (w'\mathcal{R}w \Rightarrow v(w', A) = 1) \\ 0 & \text{otherwise} \end{cases}$
6. $v(w, \blacklozenge A) := \begin{cases} 1 & \exists w' \in W : (w'\mathcal{R}w \text{ and } v(w', A) = 1) \\ 0 & \text{otherwise} \end{cases}$
7. $v(w, \neg A) := 1 - v(w, A)$
8. $v(w, A \wedge B) := \min(v(w, A), v(w, B))$
9. $v(w, A \vee B) := \max(v(w, A), v(w, B))$
10. $v(w, A \to B) := v(w, \neg A \vee B)$
11. $v(w, A \leftrightarrow B) := \begin{cases} 1 & v(w, A) = v(w, B) \\ 0 & \text{otherwise} \end{cases}$

If $\langle W, \mathcal{R}, v\rangle$ is a K_t model, then:

$$\mathsf{K}_t, \langle W, \mathcal{R}, v\rangle \models A \quad :\Leftrightarrow \quad \forall w \in W : v(w, A) = 1$$

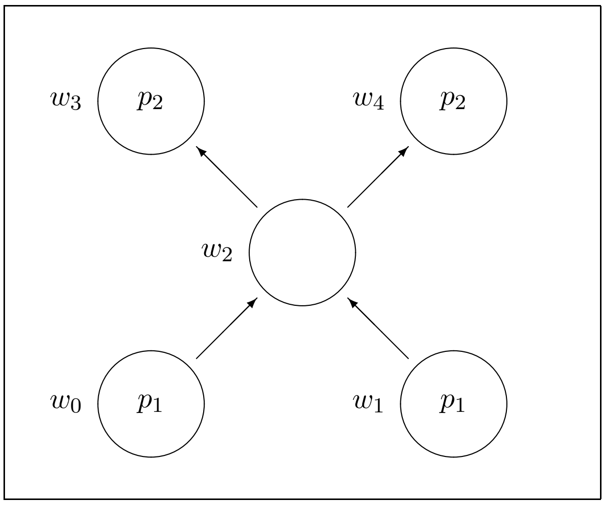


Figure 3.e: The K_t model used in the example in definition 3.8.3.

Example

In the model represented in figure 3.e we have:

- $v(w_2, \Diamond p_2) = 1,\ v(w_2, \Box p_2) = 1$
- $v(w_2, \blacksquare p_1) = 1,\ v(w_4, \blacksquare p_1) = 0,\ v(w_4, \blacksquare\blacksquare p_1) = 1$
- $v(w_3, \blacklozenge\blacklozenge(\neg p_3 \wedge \Box(\neg p_2 \vee p_3))) = 1$

3.8.4 REMARK past and future

We can interpret the accessibility relation as a time relation. Then $\Box A$ means 'in all moments in the future we will have A', and $\Diamond A$ means 'there is a moment in the future where we will have A'. $\blacksquare A$ and $\blacklozenge A$ have analogous meanings for the past: $\blacksquare A$ means 'in all moments in the past we had A', and $\blacklozenge A$ means 'there was a moment in the past where we had A'. The 'moments' must be accessible in exactly one step.

3.8.5 DEFINITION $\mathsf{K}_t \models A$

$$\mathsf{K}_t \models A \quad :\Leftrightarrow \quad \text{for all } \mathsf{K}_t \text{ models } \mathcal{M} : \mathsf{K}_t, \mathcal{M} \models A$$

If $\mathsf{K}_t \models A$, then we say that A is valid in K_t.

Example

The formula $\blacklozenge\Box p_0 \to p_0$ means: If there is a world in the past of w (accessible in exactly one step) where in each world in the future (accessible in exactly one step) p_0 holds, then p_0 holds in w. This is true in every world in every K_t model. Thus $\mathsf{K}_t \models \blacklozenge\Box p_0 \to p_0$.

3.9 OTHER LOGICS

3.9.1 REMARK S5

In contrast to S4, the accessibility relation is also symmetric, i.e. $\forall w_1, w_2 \in W : w_1 \mathcal{R} w_2$.

3.9.2 REMARK K_n, $\mathsf{K}_n + T$, KT_n, $\mathsf{KT}_n + T$, $\mathsf{S4}_n$, $\mathsf{S4}_n + T$

Instead of one accessibility relation $\mathcal{R}$ we have accessibility relations $\mathcal{R}_i$ for each $i \in \mathbb{N}$. The accessibility relation $\mathcal{R}_i$ corresponds to the connectives $\Box_i$ and $\Diamond_i$. Each accessibility relation $\mathcal{R}_i$ has to satisfy the same conditions as the accessibility relation $\mathcal{R}$ in the corresponding monomodal case, for example in the case of KT_n we have $\forall i \in \mathbb{N} : \forall w \in W : w\mathcal{R}_i w$. See for example [HM92].

3.9.3 REMARK IPC

An IPC frame is an S4 frame. An IPC model is an S4 model if $v(w, P) = 1$ and $w\mathcal{R}w'$ imply $v(w', P) = 1$ for all worlds w, w' and all variables P. In the definition of v and $\mathsf{IPC}, \mathcal{M} \models A$ we use the same definition for true, false, $\wedge$, $\vee$ as in the case of modal logics. However, $v(w, A \to B) = 1$ iff $v(w', A) = 0$ or $v(w', B) = 1$ for all worlds w' with $w\mathcal{R}w'$. Then $v(w, \neg A) := v(w, A \to \mathsf{false})$ and $v(w, A \leftrightarrow B) := v(w, (A \to B) \wedge (B \to A))$. See for example [TvD88].

3.9.4 REMARK PLTL

For PLTL there exists a possible world semantics. The frame is essentially $\mathbb{N}$ together with the usual $\leq$ relation. See for example [Eme90] for details. Note that our U is the strong until, i.e. $A\mathsf{U}C$ implies that there is a state where C is true.

3.9.5 REMARK CTL

See for example [Eme90].

3.10 SUMMARY

We have defined the usual possible world semantics for K, $\mathsf{K} + T$, KT, $\mathsf{KT} + T$, S4, $\mathsf{S4} + T$, K_t. The difference between the semantics of K and $\mathsf{K} + T$ (KT and $\mathsf{KT} + T$, S4 and $\mathsf{S4} + T$) is that we consider only those models where T is satisfied in all worlds. Note that there is an infinite number of different models for these logics. Therefore we cannot decide the validity of a formula simply by enumerating all models.

In addition, we have shown how to embed $\mathsf{S4} + T$ in S4, and we have defined the notion of the diameter of a model.

4

GRAPH CALCULI

Formally, a **graph** $\mathcal{G}$ consists of two sets, a set O of **objects** and a set A of **arrows**, and two functions $d^0, d^1 : A \to O$. Thus a graph is a "category without composition" and we will use some of the same terminology as for categories: O is the set of **objects** (or sometimes **nodes**) and A is the set of **arrows** of the graph; if f is an arrow, $d^0(f)$ is the **source** of f and $d^1(f)$ is the **target** of f.

M. Barr, C. Wells. Toposes, Triples and Theories.

4.1 INTRODUCTION

4.1.1 REMARK from possible world semantics to decision procedures

Let L be one of the logics K, $\mathsf{K}+T$, KT, $\mathsf{KT}+T$, $\mathsf{S4}$, and K_t. Starting from the possible world semantics for L, we try to find decision procedures, i.e. procedures that decide whether or not $L \models A$ for $A \in \mathrm{Fml}_L$.

We first develop so-called graph calculi. In the following chapter we will see that it is possible to obtain sequent calculi from these graph calculi.

For simplicity we consider only formulas in negation normal form in this chapter. Also most questions concerning the efficiency or termination of backward proof search are not discussed, but postponed to chapter 5, since in most cases it is easier to show the idea in the case of sequent calculi. Complexity questions are dealt with in chapter 6.

4.1.2 REMARK construct a graph calculus

For a given formula A and a logic $L \in \{\mathsf{K}, \mathsf{K}+T, \mathsf{KT}, \mathsf{KT}+T, \mathsf{S4}, \mathsf{K}_t\}$, we try to construct a countermodel, i.e. an L model $\langle W, \mathcal{R}, v\rangle$ with $v(w, A) = 0$ for some $w \in W$. If we find such a model, then obviously $L \not\models A$. If we fail after trying all possibilities, then we know that $L \models A$.

Since there is an infinite number of L models, we cannot check $L, \mathcal{M} \models \neg A$ for all L models $\mathcal{M}$. This is different for CPC, since there we have to check 'only' $2^{\text{card}(\text{vars}(A))}$ valuations for a given formula A.

Therefore we use a different procedure. We start with an L model $\langle W, \mathcal{R}, v\rangle$ and assume $v(w_0, A) = 0$ for a world $w_0 \in W$. Depending on the form of A we obtain restrictions of our model.

- If $A \equiv p_i$, then $v(w_0, p_i) = 0$.
- If $A \equiv \neg p_i$, then $v(w_0, p_i) = 1$.
- If $A \equiv B \vee C$, then $v(w_0, B) = 0$ and $v(w_0, C) = 0$.
- If $A \equiv B \wedge C$, then $v(w_0, B) = 0$ or $v(w_0, C) = 0$, i.e. we have to try two possibilities.
- If $A \equiv \Box B$, then there must exist a $w_1 \in W$ such that $w_0 \mathcal{R} w_1$ and $v(w_1, B) = 0$.
- If $A \equiv \Diamond B$, then for all $w_1 \in W$ with $w_0 \mathcal{R} w_1$ we have $v(w_1, B) = 0$.

If A is a variable or a negated variable, then we can stop. Otherwise we have conditions about its subformulas, i.e. about B and perhaps C. We reason in the same way about these formulas. If we come to a contradiction, i.e. a situation where both $v(w, p_i) = 0$ and $v(w, p_i) = 1$ for some world w and some variable p_i, then the search has failed.

Assume for example that $A \equiv \neg p_2 \vee \Box p_1$. From $v(w_0, A) = 0$ follows $v(w_0, \neg p_2) = 0$ and $v(w_0, \Box p_1) = 0$. $v(w_0, \neg p_2) = 0$ implies $v(w_0, p_2) = 1$. $v(w_0, \Box p_1) = 0$ implies the existence of a world $w_1 \in W$ with $w_0 \mathcal{R} w_1$ and $v(w_1, p_1) = 0$. Thus we end up with the three non-contradictory conditions $v(w_0, p_2) = 1$, $w_0 \mathcal{R} w_1$, and $v(w_1, p_1) = 0$.

Now assume that $A \equiv \neg \Diamond p_2 \wedge \Box p_2$. We obtain the conditions $w_0 \mathcal{R} w_1$, $v(w_1, p_2) = 0$ and $v(w_1, p_2) = 1$, which are contradictory.

This idea is realised in the following in graph calculi. An L graph can be considered as a set of conditions imposed on an L model, and a rule of a graph calculus as a rewriting rule for such conditions.

4.1.3 DEFINITION *L* graph

An L graph is a finite, connected graph with a pair of multisets of formulas of L in its vertices. Sometimes we add formulas as labels to the edges of the graph. One vertex is marked.

In the following we sometimes say 'world' if we mean a vertex of an L graph.

4.1.4 DEFINITION *L* graph scheme

An L graph scheme consists of the following parts:

- A finite, connected graph.
- A pair of schemes of multisets of formulas of L in each vertex.
- Conditions about the number of predecessors and successors of the vertices.
- A marked node (the current vertex).

An instance of an L graph scheme is an L graph that 'matches' the L graph scheme.

Example

See the premises and conclusions of the rules of the calculus $\mathsf{K}^{\mathcal{G}}$ defined in 4.2.1 for examples of L graph schemes, and the proof in figure 4.c for instances of those L graph schemes. The L graph schemes are represented according to the conventions of remark 4.1.5.

4.1.5 REMARK representation of L graph schemes

Vertices are represented as boxes, and edges as arrows between the edges. The current vertex is marked with a double frame.

- $\cdots \rightarrow$ on the left hand side of a vertex means that this vertex has exactly one predecessor (besides the predecessors shown explicitly).
- $\cdots \xrightarrow{*}$ on the left hand side of a vertex means that this vertex has an arbitrary number of predecessors (besides the predecessors shown explicitly).
- $\rightarrow \cdots$ on the right hand side of a vertex means that this vertex has exactly one successor (besides the successors shown explicitly).
- $\xrightarrow{*} \cdots$ on the right hand side of a vertex means that this vertex has an arbitrary number of successors (besides the successors shown explicitly).

4.1.6 DEFINITION graph calculus

A graph calculus for the logic L consists of the following parts:

- A finite set of axioms.

  An axiom has the form $\frac{}{G}$, where G is a graph scheme of L.
- A finite set of rules.

  A rule has the form $\frac{G_1}{G}$ or $\frac{G_1 \quad G_2}{G}$, where G_1, G_2, G are graph schemes of L. Sometimes there is an additional condition.

Instances of axioms and rules are defined 'as usual'.

4.1.7 DEFINITION $L^{\mathcal{G}} \vdash G$

Let $L^{\mathcal{G}}$ be a graph calculus for L. We inductively define $L^{\mathcal{G}} \vdash G$ for L graphs G:

- If $\frac{}{G}$ is an instance of an axiom of $L^{\mathcal{G}}$, then $L^{\mathcal{G}} \vdash G$.
- If $\frac{G_1}{G}$ is an instance of a rule of $L^{\mathcal{G}}$ and $L^{\mathcal{G}} \vdash G_1$, then $L^{\mathcal{G}} \vdash G$
- If $\frac{G_1 \quad G_2}{G}$ is an instance of a rule of $L^{\mathcal{G}}$ and $L^{\mathcal{G}} \vdash G_1$, $L^{\mathcal{G}} \vdash G_2$, then $L^{\mathcal{G}} \vdash G$.

If $L^{\mathcal{G}} \vdash G$, then we say that G is provable in $L^{\mathcal{G}}$.

4.1.8 DEFINITION depth of a proof

Let $\mathcal{P}$ be a proof of the graph G in the graph calculus $L^{\mathcal{G}}$. If $\mathcal{P}$ is an instance of an axiom, then the depth of $\mathcal{P}$ is 0. If the last step in $\mathcal{P}$ is $\frac{G_1}{G}$, then the depth of $\mathcal{P}$ is the depth of the proof of G_1 plus one. If the last step in $\mathcal{P}$ is $\frac{G_1 \quad G_2}{G}$, and d_1 resp. d_2 are the depths of the proofs of G_1 and G_2, then the depth of $\mathcal{P}$ is $1 + \max(\{d_1, d_2\})$.

4.1.9 REMARK backward proof search

We start with $\boxed{\boxed{\epsilon \mid A}}$, i.e. a graph with one vertex that contains just the formula A on the right hand side.

If $\boxed{\boxed{\epsilon \mid A}}$ is an instance of an axiom of $L^{\mathcal{G}}$, then $\boxed{\boxed{\epsilon \mid A}}$ is the whole proof. Otherwise, if $L^{\mathcal{G}} \vdash$ $\boxed{\boxed{\epsilon \mid A}}$, then $\boxed{\boxed{\epsilon \mid A}}$ must be the conclusion of an instance of a rule of $L^{\mathcal{G}}$. We check for which rules this is possible. If this is not possible for any of the rules of $L^{\mathcal{G}}$, then we know that $L^{\mathcal{G}} \nvdash$

$\boxed{\boxed{\epsilon \mid A}}$. Otherwise we compute the corresponding premises and continue with backward proof search for these premises. These premises will no longer be of the form $\boxed{\boxed{\epsilon \mid A}}$, and after some steps they may consist of several vertices. The idea of the backward proof search, however, is still the same: We first check whether it is an instance of an axiom, and if it is not such an instance, then it must be the conclusion of an instance of a rule.

4.1.10 DEFINITION (strongly) invertible rule

A rule with one premise of a graph calculus $L^{\mathcal{G}}$ is invertible iff for all instances $\frac{G_1}{G}$ of this rule:

$$L^{\mathcal{G}} \vdash G_1 \quad \Leftrightarrow \quad L^{\mathcal{G}} \vdash G$$

The rule is strongly invertible if we have in addition: If there exists a proof of G with depth d, then there exists a proof of G_1 whose depth is at most d.

A rule with two premises of a graph calculus $L^{\mathcal{G}}$ is invertible iff for all instances $\frac{G_1 \quad G_2}{G}$ of this rule:

$$L^{\mathcal{G}} \vdash G_1 \text{ and } L^{\mathcal{G}} \vdash G_2 \quad \Leftrightarrow \quad L^{\mathcal{G}} \vdash G$$

The rule is strongly invertible if we have in addition: If there exists a proof of G with depth d, then there exist proofs of G_1 and G_2 whose depth is at most d.

Note that since $\frac{G_1}{G}$ and $\frac{G_1 \quad G_2}{G}$ are instances of rules, we already know that the left hand side implies the right hand side.

4.1.11 REMARK invertible rules and efficiency

Assume that we do backward proof search in a graph calculus $L^{\mathcal{G}}$ and G is the L graph we have obtained so far. Assume further that G is not an instance of an axiom of $L^{\mathcal{G}}$. If $\frac{G'}{G}$ is an instance of an invertible rule of $L^{\mathcal{G}}$, then $L^{\mathcal{G}} \vdash G \Leftrightarrow L^{\mathcal{G}} \vdash G'$. Consequently it cannot be a mistake to apply this rule backwards. It is crucial for the efficiency of backward proof search to make use of the invertibility of rules. We use this invertibility already in this chapter since otherwise it would be almost impossible to present non-trivial examples (cp. remark 4.2.7).

However note that it is not always desirable to have a calculus with as many invertible rules as possible. For example $\mathsf{K}^{\mathcal{S}}$ (see definition 5.2.2) is better suited for backward proof search than $\mathsf{K}^{\mathcal{G}}$, although $\mathsf{K}^{\mathcal{S}}$ contains non-invertible rules and $\mathsf{K}^{\mathcal{G}}$ does not. See also the corresponding complexities in chapter 6.

4.1.12 REMARK representation of search trees

If we do backward proof search in a graph calculus, then we obtain a search tree. We use the following conventions when drawing such search trees.

- We write *fail* near a leaf if from now on only (jump) and (jump−) are applicable backwards.(These two rules are defined later on.)
- Let N be a node in the search tree with several children. If the search must be successful for all these children in order to obtain a proof for the the L graph in N, then there is a horizontal line above N.
- Let N be a node in the search tree with several children. If it is sufficient that the search is successful for one of these children in order to obtain a proof for the the L graph in N, then there is no horizontal line above N.

Example

The node 12 of the search tree in figure 4.b is marked with a horizontal line. There is no such line above the node 8 of the search tree in figure 4.j.

4.1.13 REMARK graph calculi vs. prefixed tableaux

Prefixed tableaux (see for example [Fit93]) can be seen as a notational variant of our graph calculi.

Prefixed tableaux have several advantages over graph calculi:

- It is definitely easier to define and handle prefixed tableaux. Especially no '$\cdots$' and '$*$' are required in the definitions of the calculi. In fact, a precise formal definition of the graph calculi would probably result in some sort of prefixed tableaux.
- An obvious data structure for a prefixed tableau could be some sort of trees, i.e. the same as for graph calculi. However, more efficient implementations seem to be possible if one uses variables in the prefixes (see for example [BG97]).

In spite of these advantages of prefixed tableaux, we use graph calculi in this chapter for the following reasons:

- Graph calculi are more intuitive. For example, we really see that we add a world to the current graph during the application of the ($\Box$) rule during backward proof search.
- In our view, it is easier to motivate the development of sequent calculi from graph calculi than from prefixed tableaux.

4.2 K

4.2.1 DEFINITION graph calculus $\mathsf{K}^{\mathcal{G}}$

axioms:

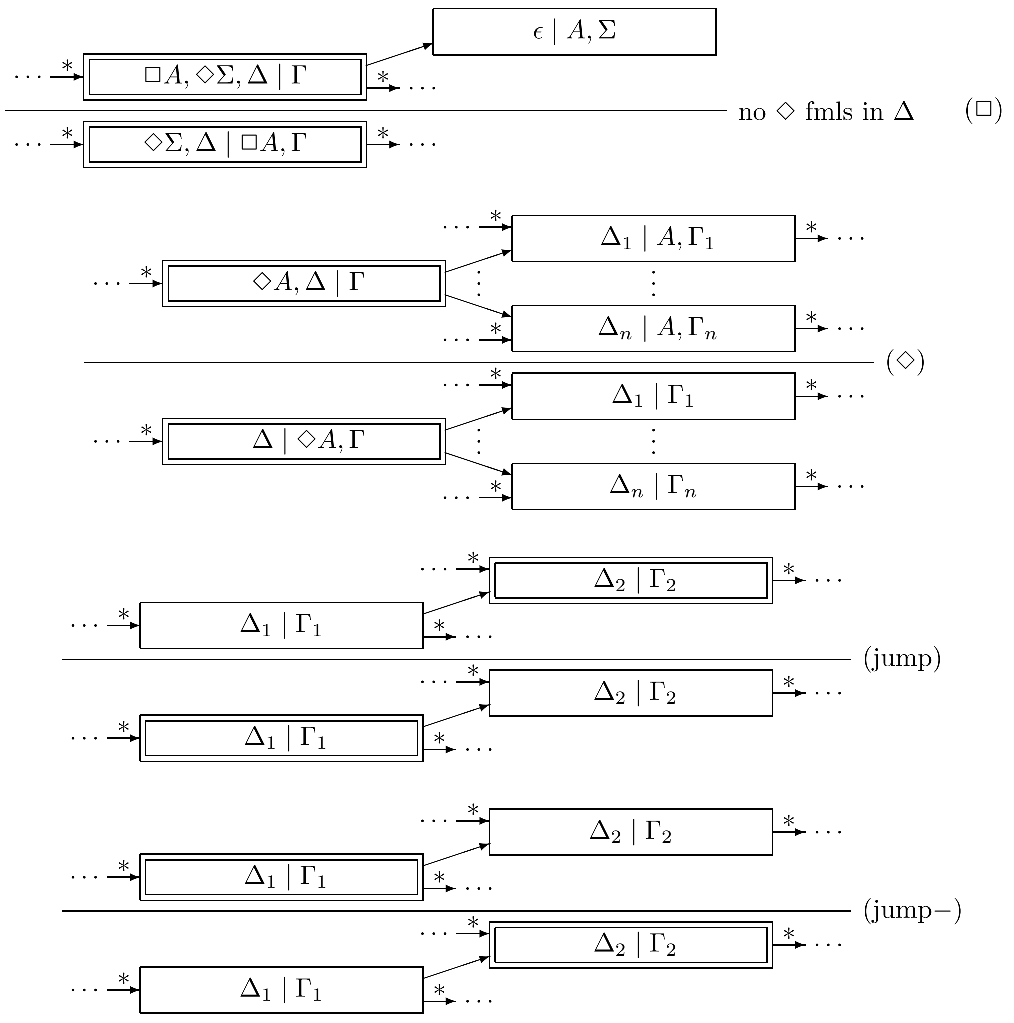
$$\frac{}{\cdots \xrightarrow{*} \boxed{\Delta \mid \mathsf{true}, \Gamma} \xrightarrow{*} \cdots}\ (\mathsf{true})$$

$$\frac{}{\cdots \xrightarrow{*} \boxed{\Delta \mid P, \neg P, \Gamma} \xrightarrow{*} \cdots}\ (\mathrm{id})$$

rules:

$$\frac{\cdots \xrightarrow{*} \boxed{A \lor B, \Delta \mid A, B, \Gamma} \xrightarrow{*} \cdots}{\cdots \xrightarrow{*} \boxed{\Delta \mid A \lor B, \Gamma} \xrightarrow{*} \cdots}\ (\lor)$$

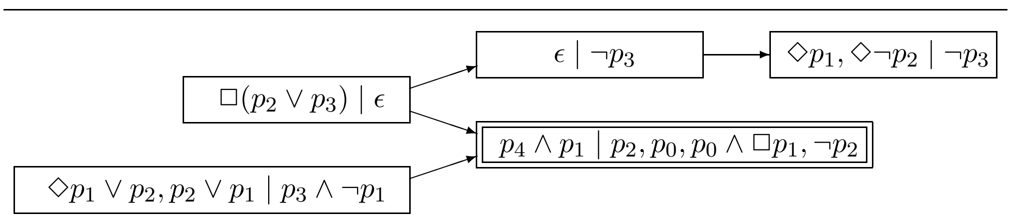
$$\frac{\cdots \xrightarrow{*} \boxed{A \land B, \Delta \mid A, \Gamma} \xrightarrow{*} \cdots \qquad \cdots \xrightarrow{*} \boxed{A \land B, \Delta \mid B, \Gamma} \xrightarrow{*} \cdots}{\cdots \xrightarrow{*} \boxed{\Delta \mid A \land B, \Gamma} \xrightarrow{*} \cdots}\ (\land)$$


$\epsilon \mid A, \Sigma$
$\Box A, \Diamond\Sigma, \Delta \mid \Gamma$
no $\Diamond$ fmls in Δ
$(\Box)$
$\Diamond\Sigma, \Delta \mid \Box A, \Gamma$
$\Delta_1 \mid A, \Gamma_1$
$\Diamond A, \Delta \mid \Gamma$
$\Delta_n \mid A, \Gamma_n$
$(\Diamond)$
$\Delta_1 \mid \Gamma_1$
$\Delta \mid \Diamond A, \Gamma$
$\Delta_n \mid \Gamma_n$
$\Delta_2 \mid \Gamma_2$
$\Delta_1 \mid \Gamma_1$
(jump)
(jump$-$)
$*$
$\cdots$

main formulas: true in (true), P and $\neg P$ in (id), $A \vee B$ in $(\vee)$, $A \wedge B$ in $(\wedge)$, $\Box A$ in $(\Box)$, $\Diamond A$ in $(\Diamond)$, none in (jump) and (jump$-$)

Example

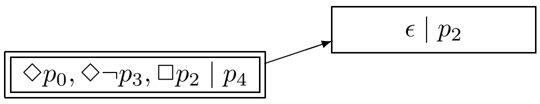
The K graph


$\epsilon \mid \neg p_3$
$\Diamond p_1, \Diamond\neg p_2 \mid \neg p_3$
$\Box(p_2 \vee p_3) \mid \epsilon$
$p_4 \wedge p_1 \mid p_2, p_0, p_0 \wedge \Box p_1, \neg p_2$
$\Diamond p_1 \vee p_2, p_2 \vee p_1 \mid p_3 \wedge \neg p_1$

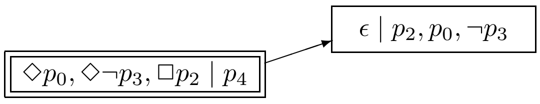
is an instance of the axiom (id).

See figure 4.c for an example of a proof in the calculus $\mathsf{K}^{\mathcal{G}}$ (and thus for instances of axioms and rules of $\mathsf{K}^{\mathcal{G}}$).

In order to motivate the $(\Box)$ rule, we assume that we start with the vertex $\boxed{\boxed{\Diamond p_0, \Diamond\neg p_3 \mid \Box p_2, p_4}}$ and want to make all formulas in it false. Using the notation from remark 4.1.2 we thus want $v(w_0, \Diamond p_0) = 0$, $v(w_0, \Diamond\neg p_3) = 0$, and $v(w_0, \Box p_2) = 0$. $v(w_0, \Box p_2) = 0$ implies that there exists a world w_1 with $w_0 \mathcal{R} w_1$ such that $v(w_1, p_2) = 0$. Thus we add a vertex and obtain

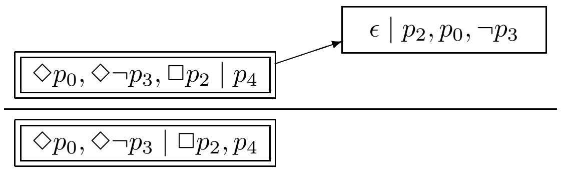


From $v(w_0, \Diamond p_0) = 0$ follows $v(w, p_0) = 0$ for all worlds w accessible from w_0. Consequently $v(w_1, p_0) = 0$, i.e. we put p_0 into the new vertex. For the same reason we also add $\neg p_3$ and obtain



$\epsilon \mid p_2, p_0, \neg p_3$
$\Diamond p_0, \Diamond\neg p_3, \Box p_2 \mid p_4$

Appending a vertex and putting p_2, p_0, p_3 in it is exactly what we would have done if we had applied the ($\Box$) rule of the calculus $\mathsf{K}^{\mathcal{G}}$ backwards on the vertex we started with, since



$\epsilon \mid p_2, p_0, \neg p_3$
$\Diamond p_0, \Diamond\neg p_3, \Box p_2 \mid p_4$
$\Diamond p_0, \Diamond\neg p_3 \mid \Box p_2, p_4$

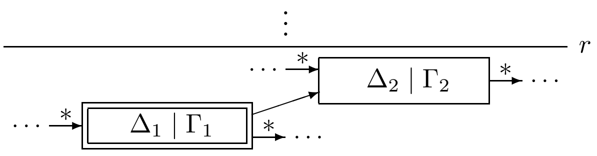
is an instance of this rule. The formulas on the left hand side are then those formulas we already dealt with.

4.2.2 THEOREM $\mathsf{K}^{\mathcal{G}}$: invertible rules

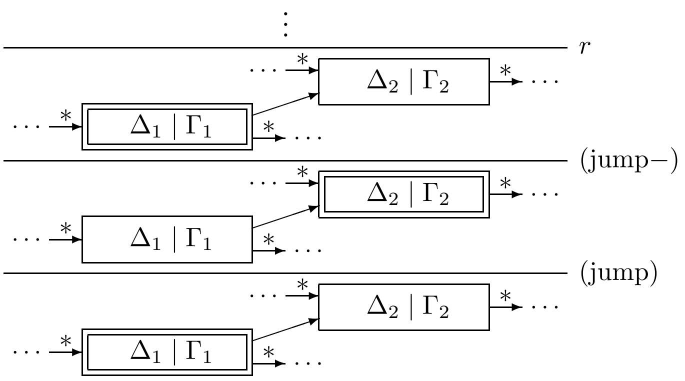
All rules of $\mathsf{K}^{\mathcal{G}}$ are invertible.

Proof

We proceed as in the proof of theorem 5.2.4 and use the notation from 5.1.10. We begin with the rules (jump) and (jump−), which are special cases. If



is a proof in $\mathsf{K}^{\mathcal{G}}$, where r is an arbitrary axiom or rule of $\mathsf{K}^{\mathcal{G}}$, then also



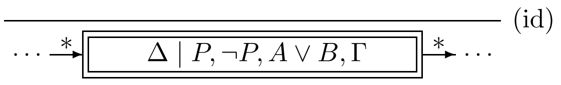
is a proof. Thus (jump) is invertible. In the same way we prove the invertibility of (jump−).

Let now r be one of the rules ($\vee$), ($\wedge$), ($\Box$), ($\Diamond$) of $\mathsf{K}^{\mathcal{G}}$. We can easily check:

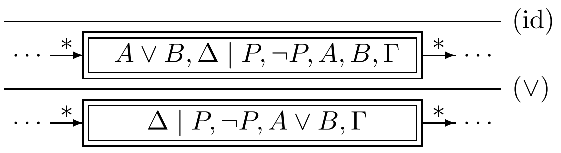
1. $(\mathsf{true}) \rightsquigarrow r(\mathsf{true})$

2. (id) $\rightsquigarrow r$(id)
3. $(\vee)r \rightsquigarrow r(\vee)$, $(\wedge)r \rightsquigarrow r(\wedge)$, $(\Box)r \rightsquigarrow r(\Box)$, $(\Diamond)r \rightsquigarrow r(\Diamond)$

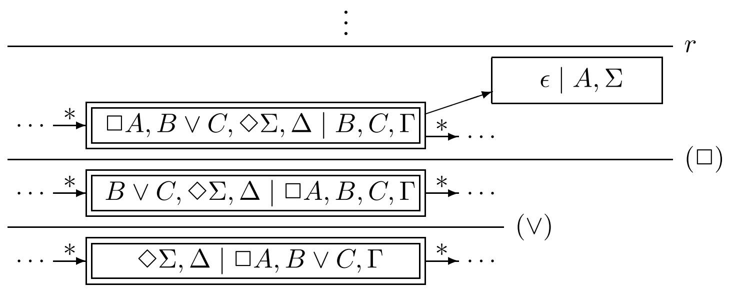
We only show two typical cases. Assume that


(id)
$\cdots \xrightarrow{*}$ $\Delta \mid P, \neg P, A \vee B, \Gamma$ $\xrightarrow{*} \cdots$

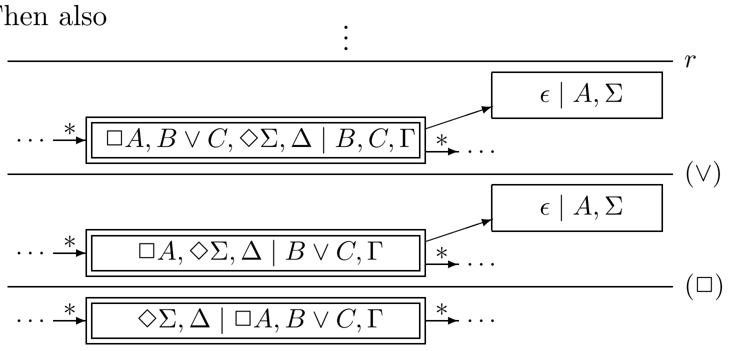
is a proof in $\mathsf{K}^{\mathcal{G}}$. Then also


(id)
$\cdots \xrightarrow{*}$ $A \vee B, \Delta \mid P, \neg P, A, B, \Gamma$ $\xrightarrow{*} \cdots$
$(\vee)$
$\cdots \xrightarrow{*}$ $\Delta \mid P, \neg P, A \vee B, \Gamma$ $\xrightarrow{*} \cdots$

is a proof in $\mathsf{K}^{\mathcal{G}}$. Thus (id) $\rightsquigarrow (\vee)$(id). Now assume that


r
$\epsilon \mid A, \Sigma$
$\cdots \xrightarrow{*}$ $\Box A, B \vee C, \Diamond\Sigma, \Delta \mid B, C, \Gamma$ $\xrightarrow{*} \cdots$
$(\Box)$
$\cdots \xrightarrow{*}$ $B \vee C, \Diamond\Sigma, \Delta \mid \Box A, B, C, \Gamma$ $\xrightarrow{*} \cdots$
$(\vee)$
$\cdots \xrightarrow{*}$ $\Diamond\Sigma, \Delta \mid \Box A, B \vee C, \Gamma$ $\xrightarrow{*} \cdots$

is a proof in $\mathsf{K}^{\mathcal{G}}$. Then also


r
$\epsilon \mid A, \Sigma$
$\cdots \xrightarrow{*}$ $\Box A, B \vee C, \Diamond\Sigma, \Delta \mid B, C, \Gamma$ $\xrightarrow{*} \cdots$
$(\vee)$
$\epsilon \mid A, \Sigma$
$\cdots \xrightarrow{*}$ $\Box A, \Diamond\Sigma, \Delta \mid B \vee C, \Gamma$ $\xrightarrow{*} \cdots$
$(\Box)$
$\cdots \xrightarrow{*}$ $\Diamond\Sigma, \Delta \mid \Box A, B \vee C, \Gamma$ $\xrightarrow{*} \cdots$

is a proof in $\mathsf{K}^{\mathcal{G}}$. Thus $(\vee)(\Box)\cdots \rightsquigarrow (\Box)(\vee)\cdots$. Finally we obtain $(\Diamond)(\Box)\cdots \rightsquigarrow (\Box)(\Diamond)\cdots$ with the transformation in figure 4.a.

Assume that we have a proof $\mathcal{P}$ of a K graph G with an application of (jump) or (jump$-$) as its last step. The rule r, which is one of $(\wedge)$, $(\Box)$, $(\Diamond)$, is applicable backwards on G (in world w and with main formula A). For each branch in this proof we assume that if we go from G upwards, then r (in world w and with main formula A) is applied backwards as soon as possible. If r (in world w and with main formula A) is not applied on a branch, then we add such an application plus the necessary applications of (jump) and (jump$-$). Afterwards we can convert this proof into a proof of G where the last step is this application of r.

From 1., 2., 3. and the transformation described above we obtain the invertibility of the rules $(\wedge)$, $(\Box)$, $(\Diamond)$.

4.2.3 REMARK termination of backward proof search

In general backward proof search in $\mathsf{K}^{\mathcal{G}}$ does not terminate since the rules (jump) and (jump$-$) are always applicable. In the following theorem we show how this can be remedied.

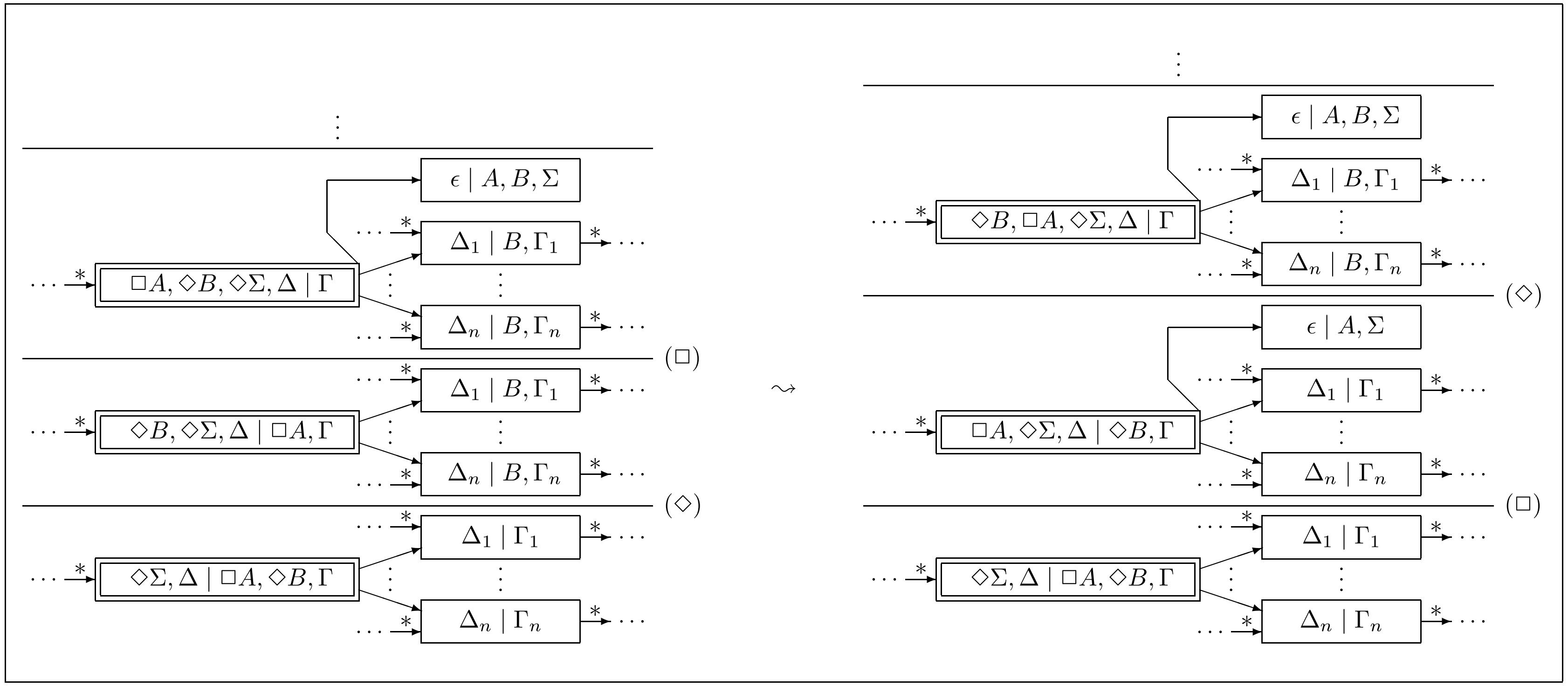

$\epsilon \mid A, B, \Sigma$
$\Delta_1 \mid B, \Gamma_1$
$\Delta_n \mid B, \Gamma_n$
$\Box A, \Diamond B, \Diamond \Sigma, \Delta \mid \Gamma$
$\Diamond B, \Diamond \Sigma, \Delta \mid \Box A, \Gamma$
$\Diamond \Sigma, \Delta \mid \Box A, \Diamond B, \Gamma$
$\Delta_1 \mid \Gamma_1$
$\Delta_n \mid \Gamma_n$
$(\Box)$
$(\Diamond)$
$\rightsquigarrow$
$\Diamond B, \Box A, \Diamond \Sigma, \Delta \mid \Gamma$
$\epsilon \mid A, \Sigma$
$\Box A, \Diamond \Sigma, \Delta \mid \Diamond B, \Gamma$
$(\Diamond)$
$(\Box)$

Figure 4.a: The transformation $(\Diamond)(\Box)\cdots \rightsquigarrow (\Box)(\Diamond)\cdots$ used in the proof of theorem 4.2.2.

4.2.4 THEOREM $\mathsf{K}^{\mathcal{G}}$: termination

Backward proof search in $\mathsf{K}^{\mathcal{G}}$ terminates if we add the following two restrictions:

- (jump) is only applied backwards if, perhaps after further backward applications of (jump), a rule different from (jump) and (jump−) is applicable backwards.
- (jump−) is never applied backwards right after (jump).

These restrictions do not change the set of provable K graphs $\boxed{\boxed{\epsilon \mid A}}$ and all rules remain invertible.

Proof

The two restrictions only forbid proofs where an application of (jump−) is followed immediately by an application of (jump). Therefore we can still prove the same K graphs $\boxed{\boxed{\epsilon \mid A}}$. From the point of view of termination the first restriction is superfluous, but without this restriction the (jump) rule would not be invertible.

Let subfmls′ be the multiset variant of the function subfmls. Thus for example $\text{subfmls}'(\Box p_0 \to \Box p_0 \vee p_1) = \Box p_0 \to \Box p_0 \vee p_1, \Box p_0, p_0, \Box p_0 \vee p_1, \Box p_0, p_0, p_1$, in contrast to $\text{subfmls}(\Box p_0 \to \Box p_0 \vee p_1) = \{\Box p_0 \to \Box p_0 \vee p_1, \Box p_0, p_0, \Box p_0 \vee p_1, p_1\}$.

When doing backward proof search in $\mathsf{K}^{\mathcal{G}}$, we can only obtain trees and not arbitrary graphs. Let G be such a tree. If $\boxed{\Delta \mid \Gamma}$ is a vertex of G and $\boxed{\Delta_1 \mid \Gamma_1}$ is one of its children, then $\text{subfmls}'(\Delta_1, \Gamma_1) \subseteq \text{subfmls}'(\Delta, \Gamma)$. Moreover, $\max\{\text{length}(A) \mid A \in \Delta_1, \Gamma_1\} < \max\{\text{length}(A) \mid A \in \Delta, \Gamma\}$. (This is not longer true for example for $\mathsf{K} + T$ and $\mathsf{S4}$.) Since a vertex can add formulas only to its children, we can prove with an induction that the length of the branches in G is at most $\text{length}(A)$, and that the branching degree is at most $\text{length}(A)$. Thus there is an upper limit (that depends only on A) for the number of formulas in G. Every backward application of one of the rules $(\vee)$, $(\wedge)$, $(\Box)$, $(\Diamond)$ moves a formula from the right to the left hand side of a vertex. Hence we can apply these rules backwards only a finite number of times.

It remains to show that it is not possible to apply (jump) and (jump−) infinitely often. This follows immediately from the second restriction, since we know that G is a finite tree.

4.2.5 THEOREM equivalence of provability and validity

If A is in negation normal form, then:

$$\mathsf{K}^{\mathcal{G}} \vdash \boxed{\boxed{\epsilon \mid A}} \quad \Leftrightarrow \quad \mathsf{K} \models A$$

Proof

'$\Leftarrow$':

We assume that $\mathsf{K}^{\mathcal{G}} \nvdash \boxed{\boxed{\epsilon \mid A}}$ and prove that there exists a K model $\langle W, \mathcal{R}, v\rangle$ such that $v(w_0, A) = 0$ for some $w_0 \in W$.

If $\mathsf{K}^{\mathcal{G}} \nvdash \boxed{\boxed{\epsilon \mid A}}$, and if we do backward proof search according to theorem 4.2.4, then there exists a branch in the search tree that fails. Let G be the K graph at the end of such a branch. If $\boxed{\Delta \mid \Gamma}$ is a vertex of G, then Γ contains only variables and negated variables, and there is no variable P such that both $P \in \Gamma$ and $\neg P \in \Gamma$.

We build a K model $\langle W, \mathcal{R}, v\rangle$ with the same structure as G. Thus W and $\mathcal{R}$ are determined, and we now have to define v. If $\boxed{\Delta \mid \Gamma}$ is a vertex of G, then we write $w(\Delta \mid \Gamma)$ for the corresponding world in the K model. For all worlds $w \in W$ and all variables P, we set

$$v(w(\Delta \mid \Gamma), P) := \begin{cases} 1 & P \notin \Gamma \\ 0 & P \in \Gamma \end{cases}$$

Using this construction, we obtain the model in figure 4.e from the K graph in node 13 in figure 4.d.

We now prove with an induction on length(B) that for all vertices $\boxed{\Delta \mid \Gamma}$ in G:

$$B \in \Gamma, \Delta \quad \Rightarrow \quad v(w(\Delta \mid \Gamma), B) = 0$$

Then $v(w_0, A) = 0$ follows immediately for the root w_0 of the K model, since A is an element of the multiset on the left hand side of the root.

- $B \equiv P$ or $B \equiv \neg P$:
  $v(w(\Delta \mid \Gamma), B) = 0$ follows from the definition of v.
- $B \equiv C \vee D$:
  From $B \in \Gamma, \Delta$ follows $C \in \Gamma, \Delta$ and $D \in \Gamma, \Delta$ (cp. the ($\vee$) rule). With the induction hypothesis we obtain $v(w(\Delta \mid \Gamma), C) = 0$ and $v(w(\Delta \mid \Gamma), D) = 0$. Thus $v(w(\Delta \mid \Gamma), C \vee D) = 0$.
- $B \equiv C \wedge D$:
  Analogous to the case $B \equiv C \vee D$.
- $B \equiv \Box C$:
  Since $\Box C \in \Gamma, \Delta$ we know that there is a successor vertex $\boxed{\Delta' \mid \Gamma'}$ of $\boxed{\Delta \mid \Gamma}$ in G such that $C \in \Gamma', \Delta'$ (cp. the rule ($\Box$)). With the induction hypothesis follows $v(w(\Delta' \mid \Gamma'), C) = 0$. Because $w(\Delta \mid \Gamma) \, \mathcal{R} \, w(\Delta' \mid \Gamma')$ we obtain $v(w(\Delta \mid \Gamma), \Box C) = 0$.
- $B \equiv \Diamond C$:
  Since $\Diamond C \in \Gamma, \Delta$ we know that in all successor vertices $\boxed{\Delta' \mid \Gamma'}$ of $\boxed{\Delta \mid \Gamma}$ in G we have $C \in \Gamma', \Delta'$ (cp. the rules ($\Diamond$) and ($\Box$)). With the induction hypothesis follows $v(w(\Delta' \mid \Gamma'), C) = 0$ for all these successor vertices. Thus $v(w(\Delta \mid \Gamma), \Diamond C) = 0$.

'$\Rightarrow$':

Let $\mathcal{P}$ be a proof of $\boxed{\boxed{\epsilon \mid A}}$ in $\mathsf{K}^{\mathcal{G}}$. Note that all K graphs that can occur in $\mathcal{P}$ are trees. First we define a translation f of K graphs that are trees into conditions about a K model $\langle W, \mathcal{R}, v \rangle$.

If G consists of a root $\boxed{\Delta \mid \Gamma}$ and n children $G_1, \ldots, G_n$, then $f(G, w_{\langle l_1,\ldots,l_m \rangle})$ is the condition

$\forall B \in \Gamma, \Delta : v(w_{\langle l_1,\ldots,l_m \rangle}, B) = 0$
and $\exists w_{\langle l_1,\ldots,l_m,1 \rangle} \in W : (w_{\langle l_1,\ldots,l_m \rangle} \mathcal{R} w_{\langle l_1,\ldots,l_m,1 \rangle}$ and $f(G_1, w_{\langle l_1,\ldots,l_m,1 \rangle}))$
and ...
and $\exists w_{\langle l_1,\ldots,l_m,n \rangle} \in W : (w_{\langle l_1,\ldots,l_m \rangle} \mathcal{R} w_{\langle l_1,\ldots,l_m,n \rangle}$ and $f(G_n, w_{\langle l_1,\ldots,l_m,n \rangle}))$

and $f(G)$ is the condition $\exists \langle W, \mathcal{R}, v \rangle : \exists w_{\langle 0 \rangle} \in W : f(G, w_{\langle 0 \rangle})$ (see also the example below).

If G is an instance of a (true) axiom of $\mathsf{K}^{\mathcal{G}}$, then the condition $f(G)$ is contradictory (in the meta-logic), since $v(w, \mathsf{true}) = 1$ for all w. $f(G)$ is also contradictory if G is an instance of an (id) axiom, since either $v(w, P) = 1$ or $v(w, \neg P) = 1$.

If $\frac{G_1}{G_3}$ is an instance of one of the rules ($\vee$), ($\Diamond$) and $f(G_1)$ is contradictory, then $f(G_3)$ is contradictory too. If $\frac{G_1 \quad G_2}{G_3}$ is an instance of the rule ($\wedge$) and both $f(G_1)$ and $f(G_2)$ are contradictory, then $f(G_3)$ is contradictory too. Applying (jump) or (jump$-$) does not change the condition.

We only show the case of the ($\Box$) rule in more detail. We assume that $\frac{G_1}{G_3}$ is an instance of the ($\Box$) rule and that $f(G_1)$ is contradictory. Thus G_1 is a K graph of the form

$$\begin{tikzcd}
 & & \boxed{\epsilon \mid A, \Sigma} \\
\cdots \arrow[r, "*"] & \boxed{\boxed{\Box A, \Diamond\Sigma, \Delta \mid \Gamma}} \arrow[ur] \arrow[r, "*"] & \cdots
\end{tikzcd}$$

and the K graph G_3 is of the form

$$\begin{tikzcd}
\cdots \arrow[r, "*"] & \boxed{\boxed{\Diamond\Sigma, \Delta \mid \Box A, \Gamma}} \arrow[r, "*"] & \cdots
\end{tikzcd}$$

Thus the condition $f(G_1)$ is of the form

$\ldots$
and $\forall B \in \Box A, \Diamond\Sigma, \Delta, \Gamma : v(w_{\langle l_1,\ldots,l_m\rangle}, B) = 0$
and $\ldots$
and $\exists w_{\langle l_1,\ldots,l_m,k\rangle} \in W : (w_{\langle l_1,\ldots,l_m\rangle} \mathcal{R} w_{\langle l_1,\ldots,l_m,k\rangle}$
and $\forall B \in A, \Sigma : v(w_{\langle l_1,\ldots,l_m,k\rangle}, B) = 0)$
and $\ldots$

and the condition $f(G_3)$ is of the form

$\ldots$
and $\forall B \in \Box A, \Diamond\Sigma, \Delta, \Gamma : v(w_{\langle l_1,\ldots,l_m\rangle}, B) = 0$
and $\ldots$

The parts of the conditions abbreviated with '$\ldots$' are both times the same and concern the vertices that are abbreviated with '$\ldots$' in G_1 and G_3.

We assume $v(w_{\langle l_1,\ldots,l_m\rangle}, \Box A) = 0$ and $v(w_{\langle l_1,\ldots,l_m\rangle}, \Diamond\Sigma) = 0$. This implies that there exists a $w_{\langle l_1,\ldots,l_m,k\rangle}$ with $w_{\langle l_1,\ldots,l_m\rangle} \mathcal{R} w_{\langle l_1,\ldots,l_m,k\rangle}$, $v(w_{\langle l_1,\ldots,l_m,k\rangle}, A) = 0$, and $v(w_{\langle l_1,\ldots,l_m,k\rangle}, \Sigma) = 0$. Therefore also G_3 is contradictory.

Thus we know that $f(\boxed{\boxed{\epsilon \mid A}})$ must be contradictory. However, $f(\boxed{\boxed{\epsilon \mid A}})$ is the condition $\exists\langle W, \mathcal{R}, v\rangle : \exists w_0 \in W : v(w_0, A) = 0$. Thus $\mathsf{K} \models A$.

Example

If G is the K graph

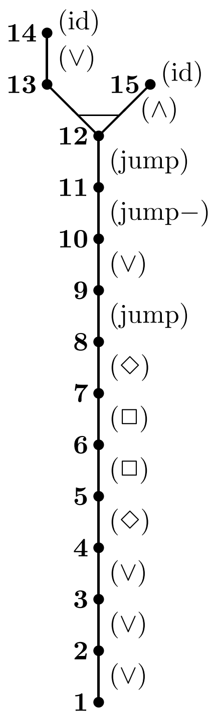
$$\begin{tikzcd}
 & \boxed{\epsilon \mid \Diamond p_3} \arrow[r] & \boxed{\epsilon \mid p_5} \\
\boxed{\boxed{\Box p_2 \mid p_1}} \arrow[ur] \arrow[dr] & & \\
 & \boxed{\epsilon \mid p_4, \neg p_6} &
\end{tikzcd}$$

then $f(G)$ is the condition

$\exists\langle W, \mathcal{R}, v\rangle : \exists w_{\langle 0\rangle} \in W :$
$(\ v(w_{\langle 0\rangle}, \Box p_2) = 0$ and $v(w_{\langle 0\rangle}, p_1) = 0$
and $\exists w_{\langle 0,1\rangle} \in W : (\ w_{\langle 0\rangle} \mathcal{R} w_{\langle 0,1\rangle}$ and $v(w_{\langle 0,1\rangle}, \Diamond p_3) = 0$
and $\exists w_{\langle 0,1,1\rangle} \in W : (w_{\langle 0,1\rangle} \mathcal{R} w_{\langle 0,1,1\rangle}$ and $v(w_{\langle 0,1,1\rangle}, p_5) = 0)\ )$
and $\exists w_{\langle 0,2\rangle} \in W : (\ w_{\langle 0\rangle} \mathcal{R} w_{\langle 0,2\rangle}$ and $v(w_{\langle 0,2\rangle}, p_4) = 0$ and $v(w_{\langle 0,2\rangle}, \neg p_6) = 0\ )\ )$

4.2.6 REMARK $\mathsf{K}^{\mathcal{G}}$: backward proof search

Assume that we do backward proof search for $\boxed{\boxed{\epsilon \mid A}}$. Since all rules are invertible, no backtracking is required during backward proof search (cp. remark 4.1.11). We use the two restrictions from theorem 4.2.4 to make sure that the search terminates (still no backtracking is required). If the search stops and one of the leaves of the search tree is not an instance of an axiom, then $\mathsf{K}^{\mathcal{G}} \not\models \boxed{\boxed{\epsilon \mid A}}$.



The search starts at node 1 with three (∨) backward applications (node 4). Then two successor worlds are created and the ◇ formulas are handled (node 8). Now we can choose between two jumps. We jump into the 'lower' world and apply (∨) backwards. No other rule is applicable in this world, so we have to jump into the other leaf. There we apply (∧) and then on one branch (∨) backwards and find two instances of axioms (nodes 14 and 15).
Thus $\boxed{\boxed{\epsilon \mid A}}$ is provable in $\mathsf{K}^{\mathcal{G}}$.
The thick lines are the parts of the search tree that belong to the proof (i.e. the search tree and the proof coincide).
The search tree (and thus the corresponding proof) would be smaller if we had jumped into the other world right after node 8.

Figure 4.b: The search tree in $\mathsf{K}^{\mathcal{G}}$ of the first example in remark 4.2.6. See figure 4.c for the corresponding proof.

Example

We check whether $\mathsf{K}^{\mathcal{G}} \vdash \boxed{\boxed{\epsilon \mid A}}$, where $A \equiv (\Diamond(\neg p_0 \vee p_3) \vee \Diamond\neg p_1) \vee (\Box p_2 \vee \Box(p_0 \wedge p_1))$.
We use the following abbreviations:

- $A_1 :\equiv \Diamond(\neg p_0 \vee p_3) \vee \Diamond\neg p_1$
- $A_2 :\equiv \Box p_2 \vee \Box(p_0 \wedge p_1)$
- $\Pi := \Diamond(\neg p_0 \vee p_3), \Diamond\neg p_1, \Box p_2, \Box(p_0 \wedge p_1), A_1, A_2, A$

Thus $A \equiv A_1 \vee A_2$. See figure 4.b for the resulting search tree and figure 4.c for the corresponding proof.

Now we do backward proof search for $\boxed{\boxed{\epsilon \mid B}}$, where $B \equiv (\Diamond p_3 \vee \Diamond\neg p_1) \vee (\Box(p_2 \vee p_0) \vee \Box(p_0 \wedge p_1))$. We use the abbreviation $\Pi = B, \Diamond p_3 \vee \Diamond\neg p_1, \Box(p_2 \vee p_0) \vee \Box(p_0 \wedge p_1), \Diamond p_3, \Diamond\neg p_1, \Box(p_2 \vee p_0), \Box(p_0 \wedge p_1)$. The proof search fails. See figure 4.d for the resulting search tree and figure 4.f for the corresponding non-proof.

In order to construct a countermodel $\mathcal{M}$, we just take the K graph 13 from figure 4.f, remove the left hand side in each node, and negate the literals (see figure 4.e). Indeed $\mathcal{M}, w_0 \models \Box\neg p_3 \wedge \Box p_1 \wedge \Diamond(\neg p_2 \wedge \neg p_0) \wedge \Diamond(\neg p_0 \vee \neg p_1)$, i.e. $\mathcal{M}, w_0 \models \neg B$.

4.2.7 REMARK effect of invertibility

If we did not know that all the rules are invertible, the search tree would become much larger: At every node we would have to try all possible ways to apply rules backwards. For example in

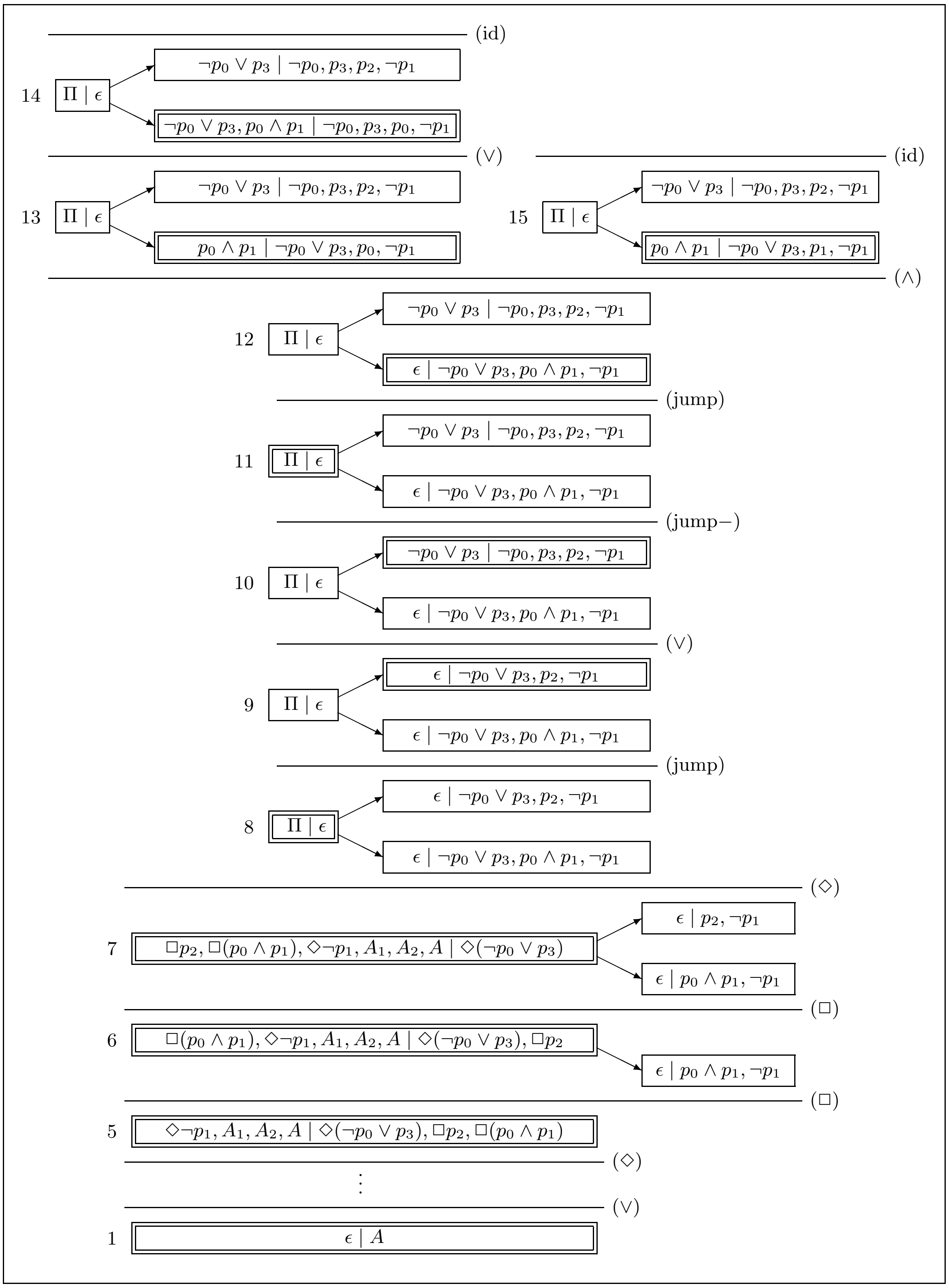
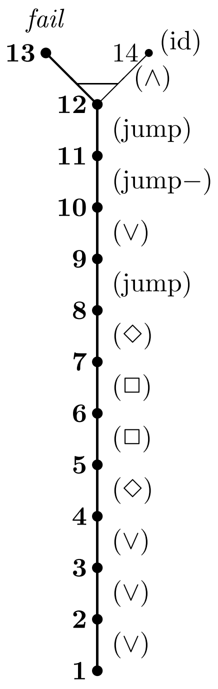

(id)
14 $\Pi \mid \epsilon$
$\neg p_0 \vee p_3 \mid \neg p_0, p_3, p_2, \neg p_1$
$\neg p_0 \vee p_3, p_0 \wedge p_1 \mid \neg p_0, p_3, p_0, \neg p_1$
($\vee$)
(id)
13 $\Pi \mid \epsilon$
$\neg p_0 \vee p_3 \mid \neg p_0, p_3, p_2, \neg p_1$
$p_0 \wedge p_1 \mid \neg p_0 \vee p_3, p_0, \neg p_1$
15 $\Pi \mid \epsilon$
$\neg p_0 \vee p_3 \mid \neg p_0, p_3, p_2, \neg p_1$
$p_0 \wedge p_1 \mid \neg p_0 \vee p_3, p_1, \neg p_1$
($\wedge$)
12 $\Pi \mid \epsilon$
$\neg p_0 \vee p_3 \mid \neg p_0, p_3, p_2, \neg p_1$
$\epsilon \mid \neg p_0 \vee p_3, p_0 \wedge p_1, \neg p_1$
(jump)
11 $\Pi \mid \epsilon$
$\neg p_0 \vee p_3 \mid \neg p_0, p_3, p_2, \neg p_1$
$\epsilon \mid \neg p_0 \vee p_3, p_0 \wedge p_1, \neg p_1$
(jump$-$)
10 $\Pi \mid \epsilon$
$\neg p_0 \vee p_3 \mid \neg p_0, p_3, p_2, \neg p_1$
$\epsilon \mid \neg p_0 \vee p_3, p_0 \wedge p_1, \neg p_1$
($\vee$)
9 $\Pi \mid \epsilon$
$\epsilon \mid \neg p_0 \vee p_3, p_2, \neg p_1$
$\epsilon \mid \neg p_0 \vee p_3, p_0 \wedge p_1, \neg p_1$
(jump)
8 $\Pi \mid \epsilon$
$\epsilon \mid \neg p_0 \vee p_3, p_2, \neg p_1$
$\epsilon \mid \neg p_0 \vee p_3, p_0 \wedge p_1, \neg p_1$
($\Diamond$)
7 $\Box p_2, \Box(p_0 \wedge p_1), \Diamond\neg p_1, A_1, A_2, A \mid \Diamond(\neg p_0 \vee p_3)$
$\epsilon \mid p_2, \neg p_1$
$\epsilon \mid p_0 \wedge p_1, \neg p_1$
($\Box$)
6 $\Box(p_0 \wedge p_1), \Diamond\neg p_1, A_1, A_2, A \mid \Diamond(\neg p_0 \vee p_3), \Box p_2$
$\epsilon \mid p_0 \wedge p_1, \neg p_1$
($\Box$)
5 $\Diamond\neg p_1, A_1, A_2, A \mid \Diamond(\neg p_0 \vee p_3), \Box p_2, \Box(p_0 \wedge p_1)$
($\Diamond$)
$\vdots$
($\vee$)
1 $\epsilon \mid A$

Figure 4.c: Proof in $\mathsf{K}^{\mathcal{G}}$ of the first example in remark 4.2.6.



The search starts at node 1 with three (∨) backward applications (node 4). Then two successor worlds are created, and the ◇ formulas are handled (node 8). Now we have to choose between two jumps. We jump into one world, and there we can apply (∨) backwards. We cannot do anything else in this world, so we jump back to the root and from there into the other leaf. Only one of the two graphs obtained after a backwards application of (∧) is an instance of an axiom (node 14). The other one (node 13) is not an instance of an axiom, but a leaf of the search tree. Thus $\boxed{\boxed{\epsilon \mid B}}$ is not provable in $\mathsf{K}^{\mathcal{G}}$.
The thick lines are the parts of the search tree that are used for the countermodel.

Figure 4.d: Search tree in $\mathsf{K}^{\mathcal{G}}$ of the second example in remark 4.2.6. See figure 4.f for the corresponding non-proof and figure 4.e for the resulting countermodel.

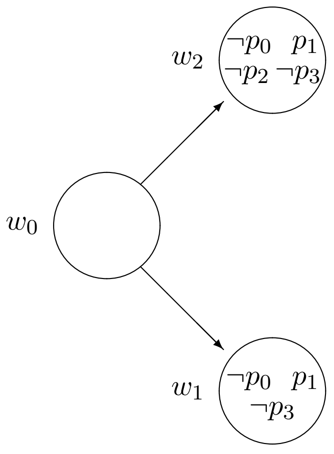


Figure 4.e: Countermodel of B of the example in remark 4.2.6. See figure 4.f for the corresponding non-proof.

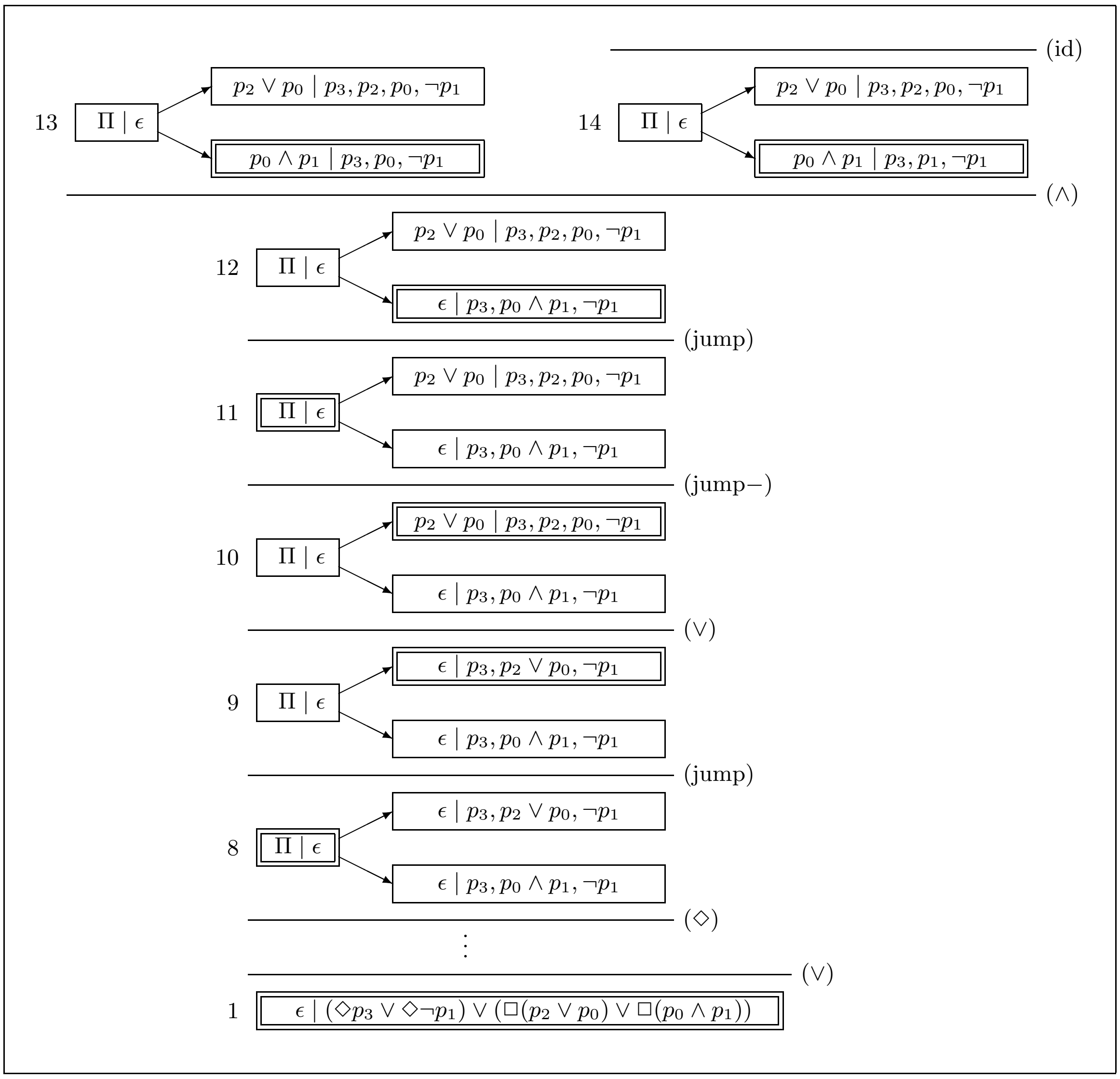

(id)
13
$\Pi \mid \epsilon$
$p_2 \vee p_0 \mid p_3, p_2, p_0, \neg p_1$
$p_0 \wedge p_1 \mid p_3, p_0, \neg p_1$
14
$\Pi \mid \epsilon$
$p_2 \vee p_0 \mid p_3, p_2, p_0, \neg p_1$
$p_0 \wedge p_1 \mid p_3, p_1, \neg p_1$
($\wedge$)
12
$\Pi \mid \epsilon$
$p_2 \vee p_0 \mid p_3, p_2, p_0, \neg p_1$
$\epsilon \mid p_3, p_0 \wedge p_1, \neg p_1$
(jump)
11
$\Pi \mid \epsilon$
$p_2 \vee p_0 \mid p_3, p_2, p_0, \neg p_1$
$\epsilon \mid p_3, p_0 \wedge p_1, \neg p_1$
(jump$-$)
10
$\Pi \mid \epsilon$
$p_2 \vee p_0 \mid p_3, p_2, p_0, \neg p_1$
$\epsilon \mid p_3, p_0 \wedge p_1, \neg p_1$
($\vee$)
9
$\Pi \mid \epsilon$
$\epsilon \mid p_3, p_2 \vee p_0, \neg p_1$
$\epsilon \mid p_3, p_0 \wedge p_1, \neg p_1$
(jump)
8
$\Pi \mid \epsilon$
$\epsilon \mid p_3, p_2 \vee p_0, \neg p_1$
$\epsilon \mid p_3, p_0 \wedge p_1, \neg p_1$
($\Diamond$)
⋮
($\vee$)
1
$\epsilon \mid (\Diamond p_3 \vee \Diamond \neg p_1) \vee (\Box(p_2 \vee p_0) \vee \Box(p_0 \wedge p_1))$

Figure 4.f: Non-proof of $\mathsf{K}^{\mathcal{G}}$ of the second example in remark 4.2.6. See figure 4.d for the corresponding search tree and figure 4.e for the resulting countermodel.

node 5 in the search tree in figure 4.b we have three possibilities: we can apply ($\Box$) backwards in two different ways, and we can apply ($\Diamond$) backwards.

If do not make use of the invertibility, then already for the relatively short formula $(\Diamond(\neg p_0 \vee p_3) \vee \Diamond\neg p_1) \vee (\Box p_2 \vee \Box(p_0 \wedge p_1))$ of the example in remark 4.2.6, the search tree becomes too large to draw it completely. Figure 4.g shows the first six levels of this search tree. The thick lines and the numbers correspond to the thick lines and numbers in the search tree in figure 4.b.

4.2.8 REMARK from $\mathsf{K}^{\mathcal{G}}$ to $\mathsf{K}^{\mathcal{G},2}$

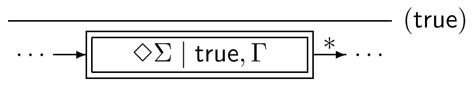
It is possible to simplify the calculus $\mathsf{K}^{\mathcal{G}}$.

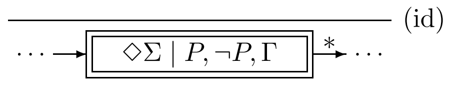
- Only trees can occur during backward proof search. Consequently we can omit many '$*$' and '$\longrightarrow$'.
- It is sufficient to put $\Diamond$ formulas on the left hand side of the sequents.
- We can do without the rule (jump−).

In this way we obtain the graph calculus $\mathsf{K}^{\mathcal{G},2}$ defined below.

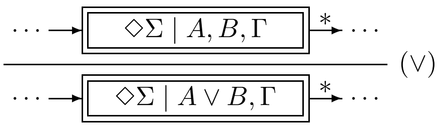
4.2.9 DEFINITION graph calculus $\mathsf{K}^{\mathcal{G},2}$

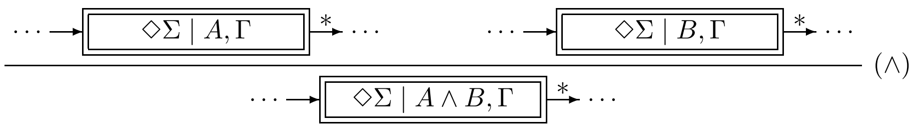
axioms:

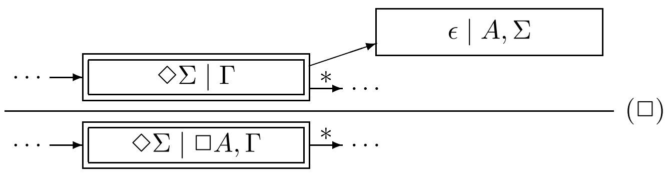


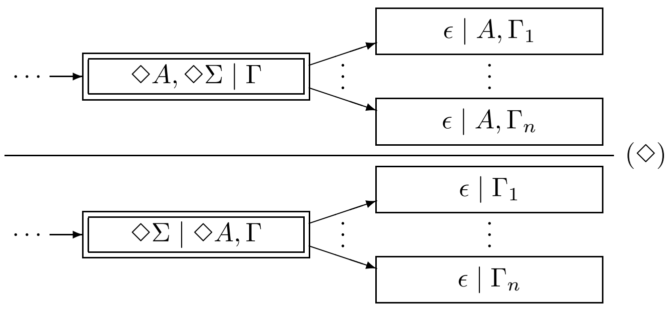


rules:









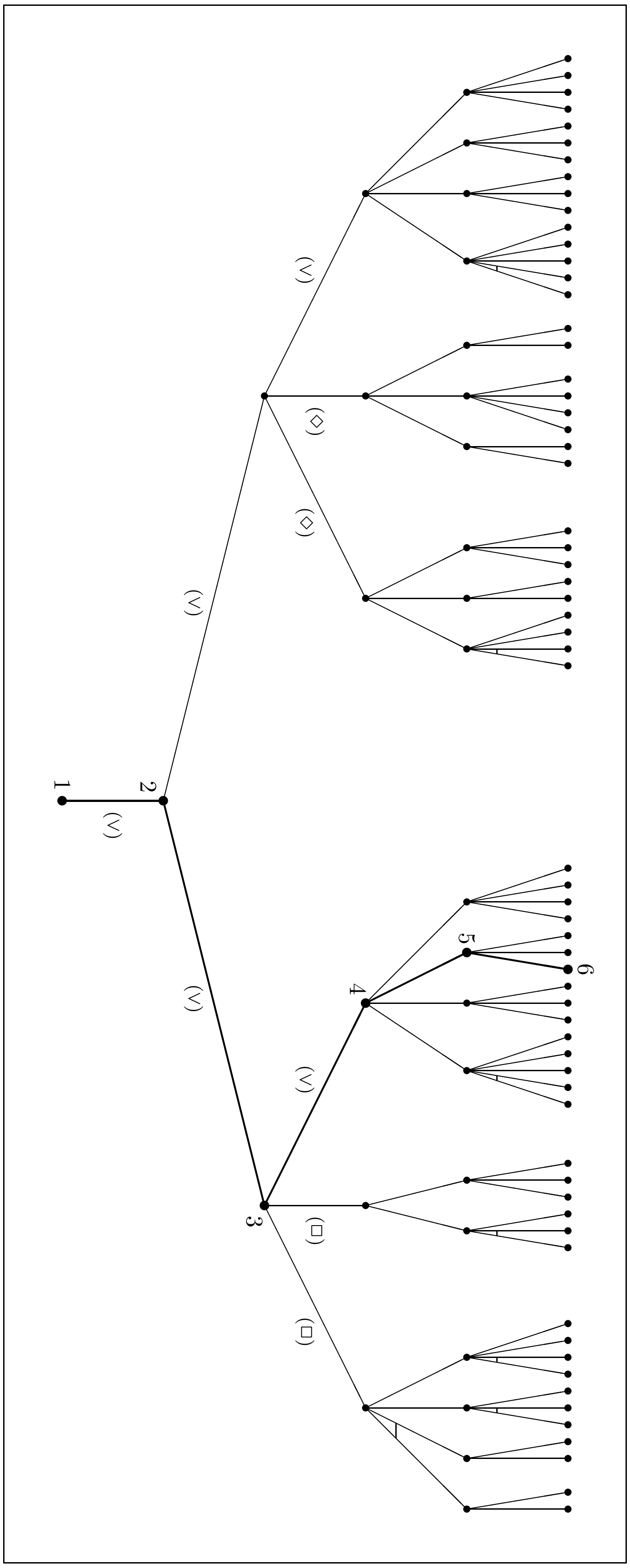


Figure 4.g: The first six levels of the complete search tree from remark 4.2.7. The thick branch and the numbers correspond to the search tree in figure 4.b.

$$\begin{tikzcd}[row sep=small] & & \boxed{\boxed{\epsilon \mid \Gamma_2}} \\ \cdots \arrow[r] & \boxed{\Diamond\Sigma_1 \mid \Gamma_1} \arrow[ur] \arrow[r, "*"] & \cdots \end{tikzcd}$$

(jump)

$$\begin{tikzcd}[row sep=small] & & \boxed{\epsilon \mid \Gamma_2} \\ \cdots \arrow[r] & \boxed{\boxed{\Diamond\Sigma_1 \mid \Gamma_1}} \arrow[ur] \arrow[r, "*"] & \cdots \end{tikzcd}$$

main formulas: true in (true), P and $\neg P$ in (id), $A \vee B$ in ($\vee$), $A \wedge B$ in ($\wedge$), $\Box A$ in ($\Box$), $\Diamond A$ in ($\Diamond$), none in (jump) and (jump$-$)

4.2.10 THEOREM $\mathsf{K}^{\mathcal{G},2}$: invertible rules

The rule (jump) of $\mathsf{K}^{\mathcal{G},2}$ is not invertible. All the other rules are strongly invertible.

Proof

Let r be one of the rules ($\vee$), ($\wedge$), ($\Box$), ($\Diamond$). We can easily check (cp. theorem 4.2.2) the following assertions:

- (true) $\rightsquigarrow r$(true)
- (id) $\rightsquigarrow r$(id)
- $(\vee)r \rightsquigarrow r(\vee)$, $(\wedge)r \rightsquigarrow r(\wedge)$, $(\Box)r \rightsquigarrow r(\Box)$, $(\Diamond)r \rightsquigarrow r(\Diamond)$
- (jump) $\rightsquigarrow r$(jump)

Thus ($\vee$), ($\wedge$), ($\Box$), ($\Diamond$) are invertible.
Since for example the K graph

$$\begin{tikzcd}[row sep=small] & \boxed{\epsilon \mid (p_0 \vee p_1) \vee \neg p_0} \\ \boxed{\boxed{\Box((p_0 \vee p_1) \vee p_0), \Box p_2 \mid \epsilon}} \arrow[ur] \arrow[dr] & \\ & \boxed{\epsilon \mid p_2} \end{tikzcd}$$

is provable in $\mathsf{K}^{\mathcal{G},2}$, but

$$\begin{tikzcd}[row sep=small] & \boxed{\epsilon \mid (p_0 \vee p_1) \vee \neg p_0} \\ \boxed{\Box((p_0 \vee p_1) \vee p_0), \Box p_2 \mid \epsilon} \arrow[ur] \arrow[dr] & \\ & \boxed{\boxed{\epsilon \mid p_2}} \end{tikzcd}$$

is not provable in $\mathsf{K}^{\mathcal{G},2}$, the rule (jump) is not invertible.

4.2.11 THEOREM equivalence of $\mathsf{K}^{\mathcal{G}}$ and $\mathsf{K}^{\mathcal{G},2}$

$$\mathsf{K}^{\mathcal{G}} \vdash \boxed{\boxed{\epsilon \mid A}} \quad\Leftrightarrow\quad \mathsf{K}^{\mathcal{G},2} \vdash \boxed{\boxed{\epsilon \mid A}}$$

Proof

‘$\Leftarrow$’:

Since $\mathsf{K}^{\mathcal{G},2}$ is essentially a restriction of $\mathsf{K}^{\mathcal{G}}$, we can use the ‘same’ proof in $\mathsf{K}^{\mathcal{G}}$.

‘$\Rightarrow$’:

Let $\mathcal{P}$ be the proof of $\boxed{\boxed{\epsilon \mid A}}$ in $\mathsf{K}^{\mathcal{G}}$.

In order to obtain a proof in $\mathsf{K}^{\mathcal{G},2}$, we have to eliminate the (jump−) applications. The proof in figure 4.h (left hand side of vertices abbreviated by '...') shows that the part of a proof between a (jump−) and a (jump) application is not always superfluous. These parts can only be eliminated without further changes of the proof if no ($\wedge$) applications occur in it.

In the following we show how $\mathcal{P}$ can be converted into a proof without (jump−) applications. Therefore we assume that there is such an application in $\mathcal{P}$.

First we assume that this application of (jump−) is followed by an application of ($\vee$), ($\Box$), or ($\Diamond$). Let X be the current world of this application. As a first step we remove this application after (jump−). Then we look at each branch b in the subproof above the (jump−) application. If on the branch b the world X is never the current world, then we do nothing. Otherwise we insert the deleted application where X is the current world for the first time (seen from the (jump−) application). Let $\mathcal{P}'$ be the result. $\mathcal{P}'$ can be adapted such that we again obtain a proof of $\boxed{\boxed{\epsilon \mid A}}$ in $\mathsf{K}^{\mathcal{G}}$. Note that the depth of the proof is not changed. If (jump−) is followed by ($\wedge$), then the procedure is a bit more complicated, but the idea remains the same.

It is crucial for this transformation that there is no rule that changes worlds that are nearer to the root than the current world. This is not the case for $\mathsf{K}_t^{\mathcal{G}}$, and therefore we cannot prove an analogous theorem for this calculus.

The proof in figure 4.h is constructed from the one in figure 4.i using this transformation.

We know that the root of the proof is a K graph with one node. Using this transformation we therefore finally obtain a proof where the (jump−) application is followed by an application of (jump). Since the K graph is a tree, we can simply remove these two applications as they cancel each other out.

4.2.12 REMARK $\mathsf{K}^{\mathcal{G},2}$: backward proof search

Since $\mathsf{K}^{\mathcal{G},2}$ contains a non-invertible rule (cp. theorem 4.2.10), backtracking is required during backward proof search. If a leaf of the search tree is not an instance of an axiom, then we backtrack until we come to a graph where we could also jump into another world.

If a formula is provable, then the proof is a subtree of the search tree with the following properties:

- The root of the search tree is the root of the proof.
- If a node of the search tree is in the proof, and this node has exactly one child, then its child is also in the proof.
- Both children of a node at a branching that is marked with a horizontal line are in the proof.
- Exactly one child of a node at a branching that is not marked with a horizontal line is in the proof.

If a formula is not provable, then we can extract a countermodel from the search tree. With respect to the search tree, a countermodel is dual to a proof: For a countermodel we take both children of branchings with horizontal lines, but only one branch of branchings without horizontal lines.

Example

We check whether $\mathsf{K}^{\mathcal{G},2} \vdash \boxed{\boxed{\epsilon \mid A}}$, where $A \equiv (\Diamond(\neg p_0 \vee p_3) \vee \Diamond\neg p_1) \vee (\Box p_2 \vee \Box(p_0 \wedge p_1))$. Compare the resulting search tree in figure 4.j with the search tree in figure 4.b.

On the left branch of the search tree we obtain the non-proof displayed in figure 4.k. The K graph at node 10 is not an instance of (id). See figure 4.l for the proof we finally obtain after backtracking (with $\Pi = \Diamond(\neg p_0 \vee p_3), \Diamond\neg p_1$).

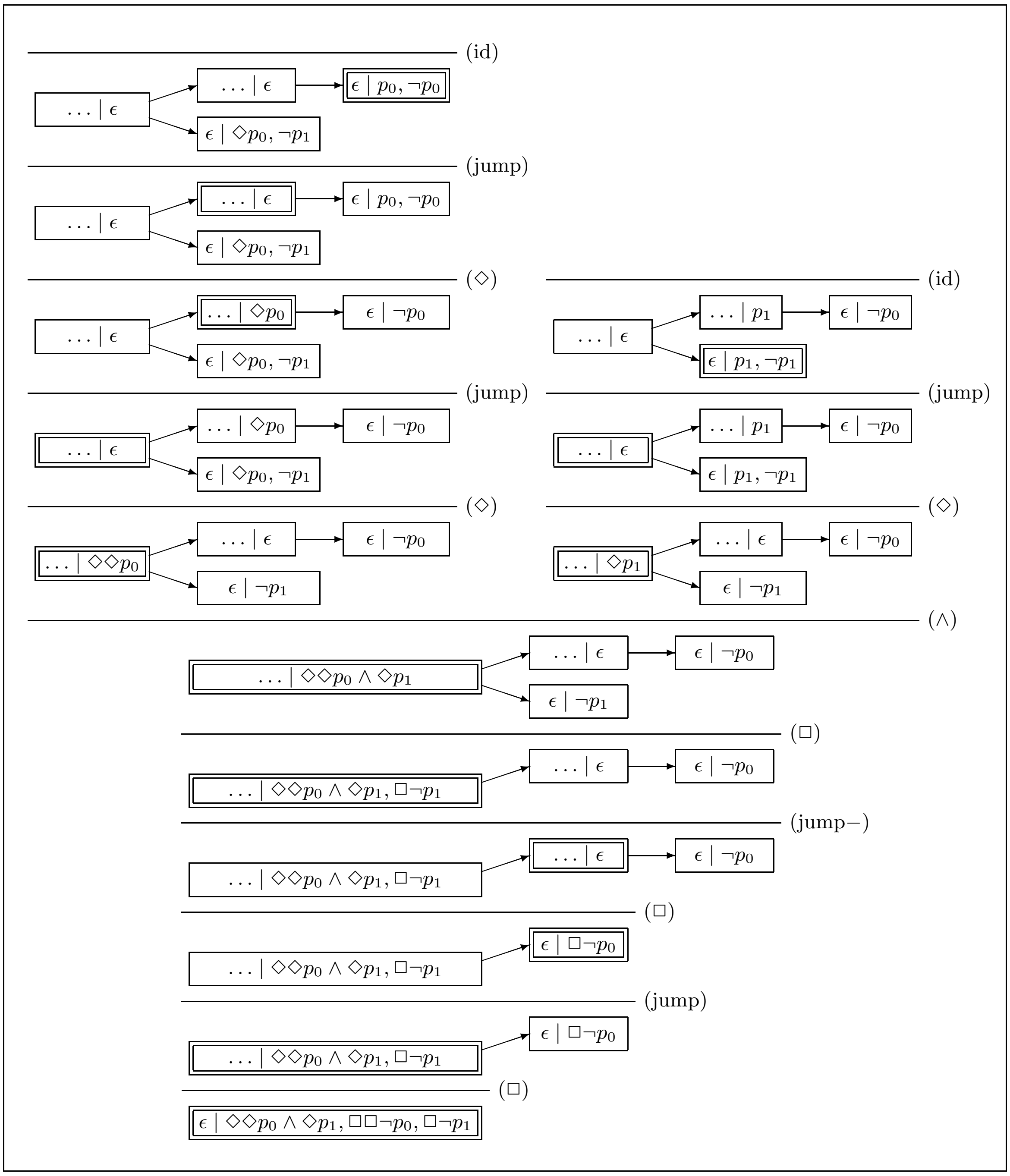


Figure 4.h: A proof in $\mathsf{K}^{\mathcal{G}}$ that contains a (jump−) application. See also the proof of theorem 4.2.11.

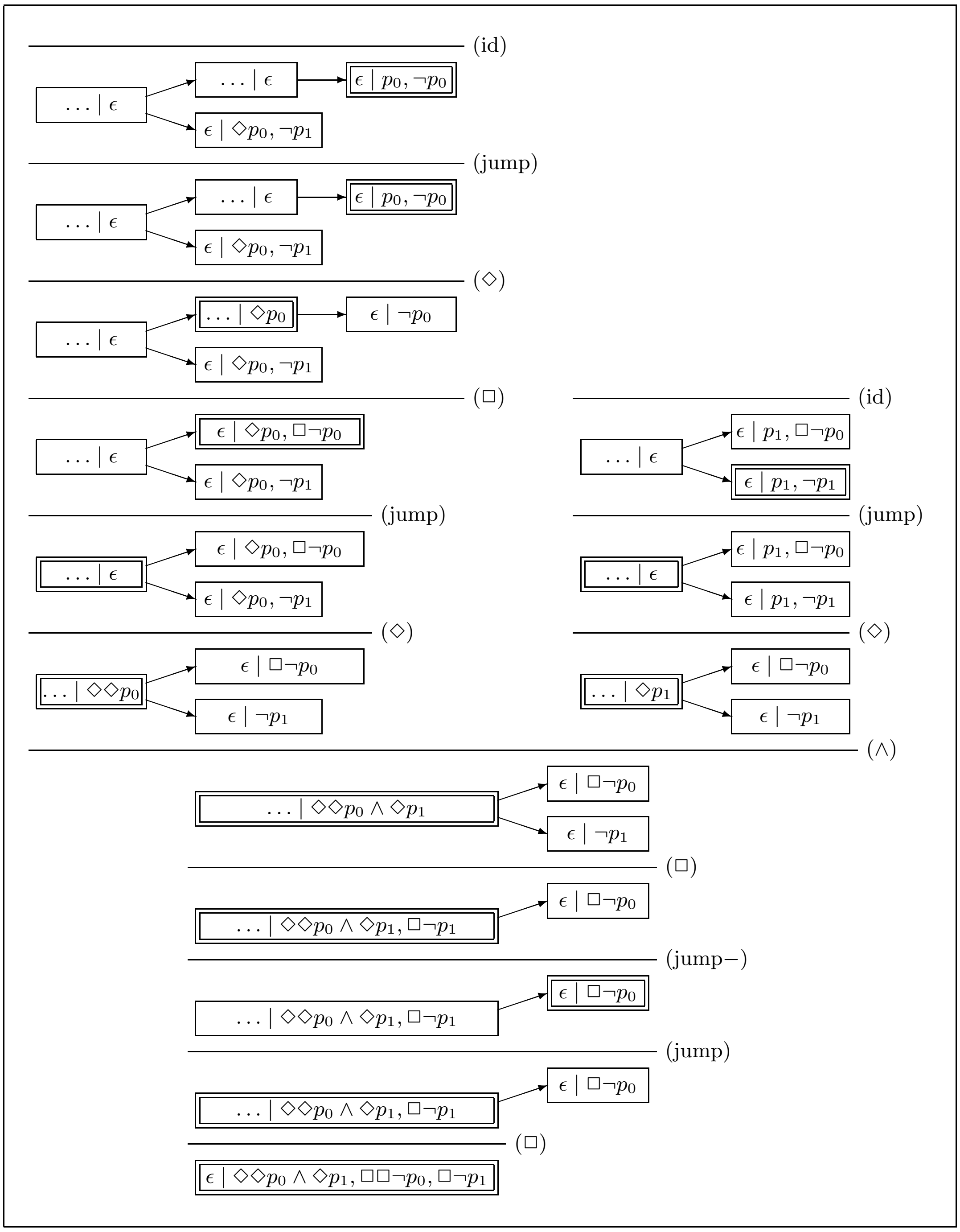
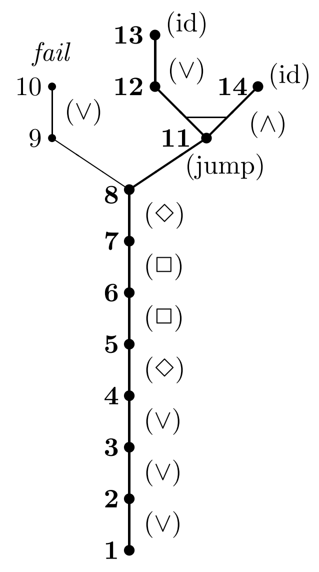


Figure 4.i: The proof of figure 4.h with the (□) application between (jump−) and (jump) moved upwards. See also the proof of theorem 4.2.11.



The search starts at node 1 with three (∨) backward applications (node 4). Then two successor worlds are created, and the ◇ formulas are handled (node 8). Now we have to choose between two jumps. One jump ends with a fail (node 10). We backtrack and try the other jump. (Note that in the search tree in figure 4.b we could apply (jump−) backwards in this situation.) This branch has another branching because of the (∧) rule and ends in two instances of the (id) axiom (nodes 13 and 14). Thus $\boxed{\boxed{\epsilon \mid A}}$ is provable in $\mathsf{K}^{\mathcal{G},2}$. The thick lines are the parts of the search tree that belong to the proof.

Figure 4.j: Search tree in $\mathsf{K}^{\mathcal{G},2}$ of the first example in remark 4.2.12. See figure 4.k for the corresponding proof.

Now we check whether $\mathsf{K}^{\mathcal{G},2} \vdash \boxed{\boxed{\epsilon \mid B}}$, where $B \equiv (\Diamond p_3 \vee \Diamond\neg p_1) \vee (\Box(p_2 \vee p_0) \vee \Box(p_0 \wedge p_1))$. Compare the resulting search tree in figure 4.m with the search tree in figure 4.d.

We set $\Pi = \Diamond p_3, \Box(p_2 \vee p_0), \Box(p_0 \wedge p_1), \Diamond\neg p_1$. The incomplete proof that ends in node 9 is displayed in 4.n. The incomplete proof that ends in the nodes 11 and 12 is displayed in figure 4.o.

The corresponding countermodel $\mathcal{M}$ has three worlds (see figure 4.p). We put the negations of the literals in the current world of node 8 into w_0, the negations of the literals in the current world of node 10 into w_1, and the negations of the literals in the current world of node 12 into w_2. Indeed $\mathcal{M}, w_0 \models \Box\neg p_3 \wedge \Box p_1 \wedge \Diamond(\neg p_2 \wedge \neg p_0) \wedge \Diamond(\neg p_0 \vee \neg p_1)$, i.e. $\mathcal{M}, w_0 \models \neg((\Diamond p_3 \vee \Diamond\neg p_1) \vee (\Box(p_2 \vee p_0) \vee \Box(p_0 \wedge p_1)))$. See chapter 5 for more examples for the extraction of a countermodel from a failed backward proof search.

4.3 $\mathsf{K}+T$

4.3.1 REMARK from $\mathsf{K}^{\mathcal{G}}$ to $(\mathsf{K}+T)^{\mathcal{G}}$

In $\mathsf{K}+T$ models, all formulas of T must be true in all worlds, i.e. their negations must be false everywhere. Therefore we have to make sure that the negations are put in all vertices during backward proof search in $(\mathsf{K}+T)^{\mathcal{G}}$. The (□) rule does it whenever a new vertex is appended, and in addition we have to do add $\mathrm{nnf}(\neg T)$ right at the beginning (cp. theorem 4.3.4).

4.3.2 DEFINITION graph calculus $(\mathsf{K}+T)^{\mathcal{G}}$

$(\mathsf{K}+T)^{\mathcal{G}}$ is the calculus $\mathsf{K}^{\mathcal{G}}$ with the (□) rule replaced by the rule

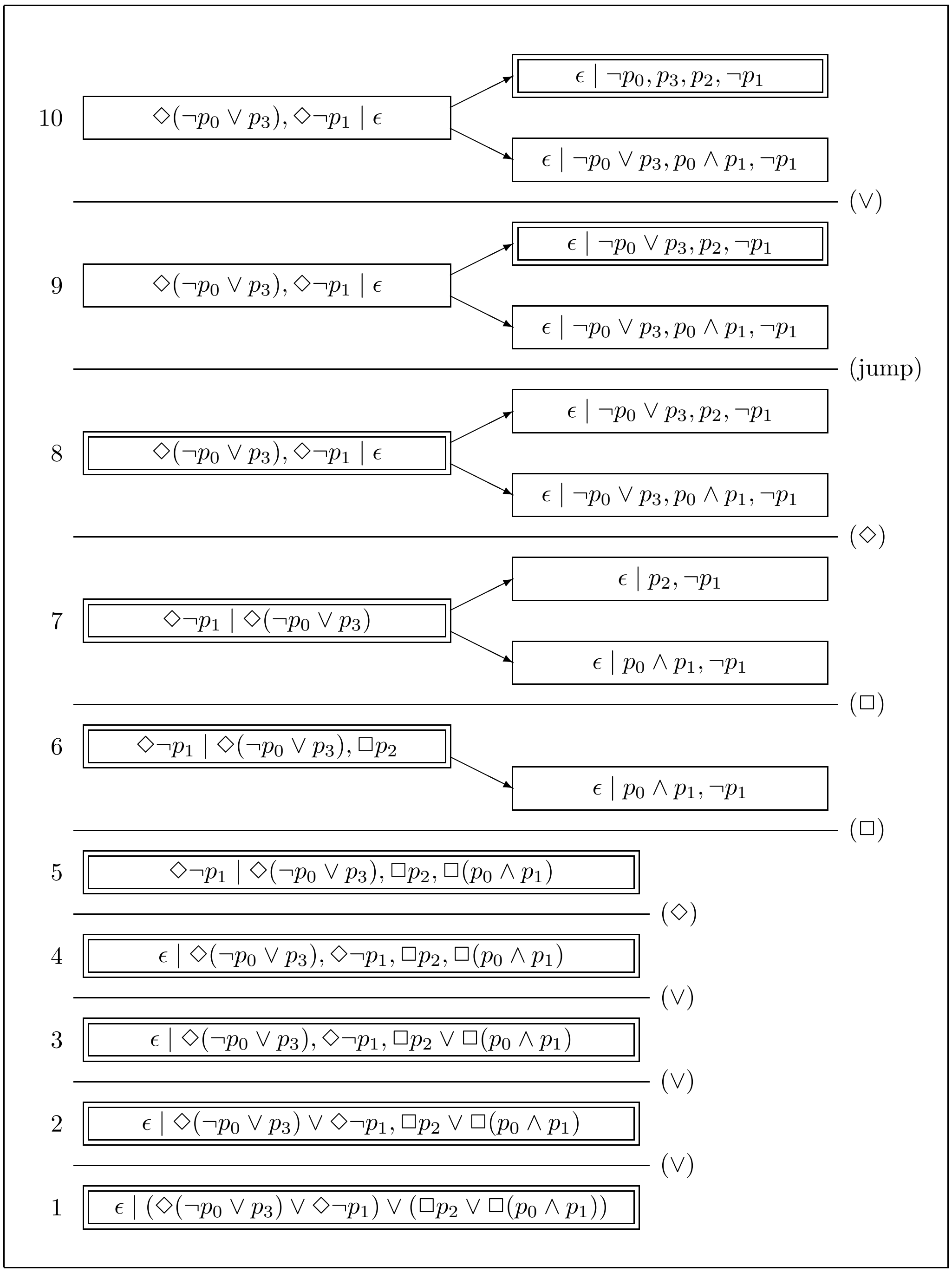

10 $\Diamond(\neg p_0 \vee p_3), \Diamond\neg p_1 \mid \epsilon$
$\epsilon \mid \neg p_0, p_3, p_2, \neg p_1$
$\epsilon \mid \neg p_0 \vee p_3, p_0 \wedge p_1, \neg p_1$
$(\vee)$
9 $\Diamond(\neg p_0 \vee p_3), \Diamond\neg p_1 \mid \epsilon$
$\epsilon \mid \neg p_0 \vee p_3, p_2, \neg p_1$
$\epsilon \mid \neg p_0 \vee p_3, p_0 \wedge p_1, \neg p_1$
(jump)
8 $\Diamond(\neg p_0 \vee p_3), \Diamond\neg p_1 \mid \epsilon$
$\epsilon \mid \neg p_0 \vee p_3, p_2, \neg p_1$
$\epsilon \mid \neg p_0 \vee p_3, p_0 \wedge p_1, \neg p_1$
$(\Diamond)$
7 $\Diamond\neg p_1 \mid \Diamond(\neg p_0 \vee p_3)$
$\epsilon \mid p_2, \neg p_1$
$\epsilon \mid p_0 \wedge p_1, \neg p_1$
$(\Box)$
6 $\Diamond\neg p_1 \mid \Diamond(\neg p_0 \vee p_3), \Box p_2$
$\epsilon \mid p_0 \wedge p_1, \neg p_1$
$(\Box)$
5 $\Diamond\neg p_1 \mid \Diamond(\neg p_0 \vee p_3), \Box p_2, \Box(p_0 \wedge p_1)$
$(\Diamond)$
4 $\epsilon \mid \Diamond(\neg p_0 \vee p_3), \Diamond\neg p_1, \Box p_2, \Box(p_0 \wedge p_1)$
$(\vee)$
3 $\epsilon \mid \Diamond(\neg p_0 \vee p_3), \Diamond\neg p_1, \Box p_2 \vee \Box(p_0 \wedge p_1)$
$(\vee)$
2 $\epsilon \mid \Diamond(\neg p_0 \vee p_3) \vee \Diamond\neg p_1, \Box p_2 \vee \Box(p_0 \wedge p_1)$
$(\vee)$
1 $\epsilon \mid (\Diamond(\neg p_0 \vee p_3) \vee \Diamond\neg p_1) \vee (\Box p_2 \vee \Box(p_0 \wedge p_1))$

Figure 4.k: Non-proof in $\mathsf{K}^{\mathcal{G},2}$ of the first example in remark 4.2.12. See figure 4.j for the corresponding search tree.

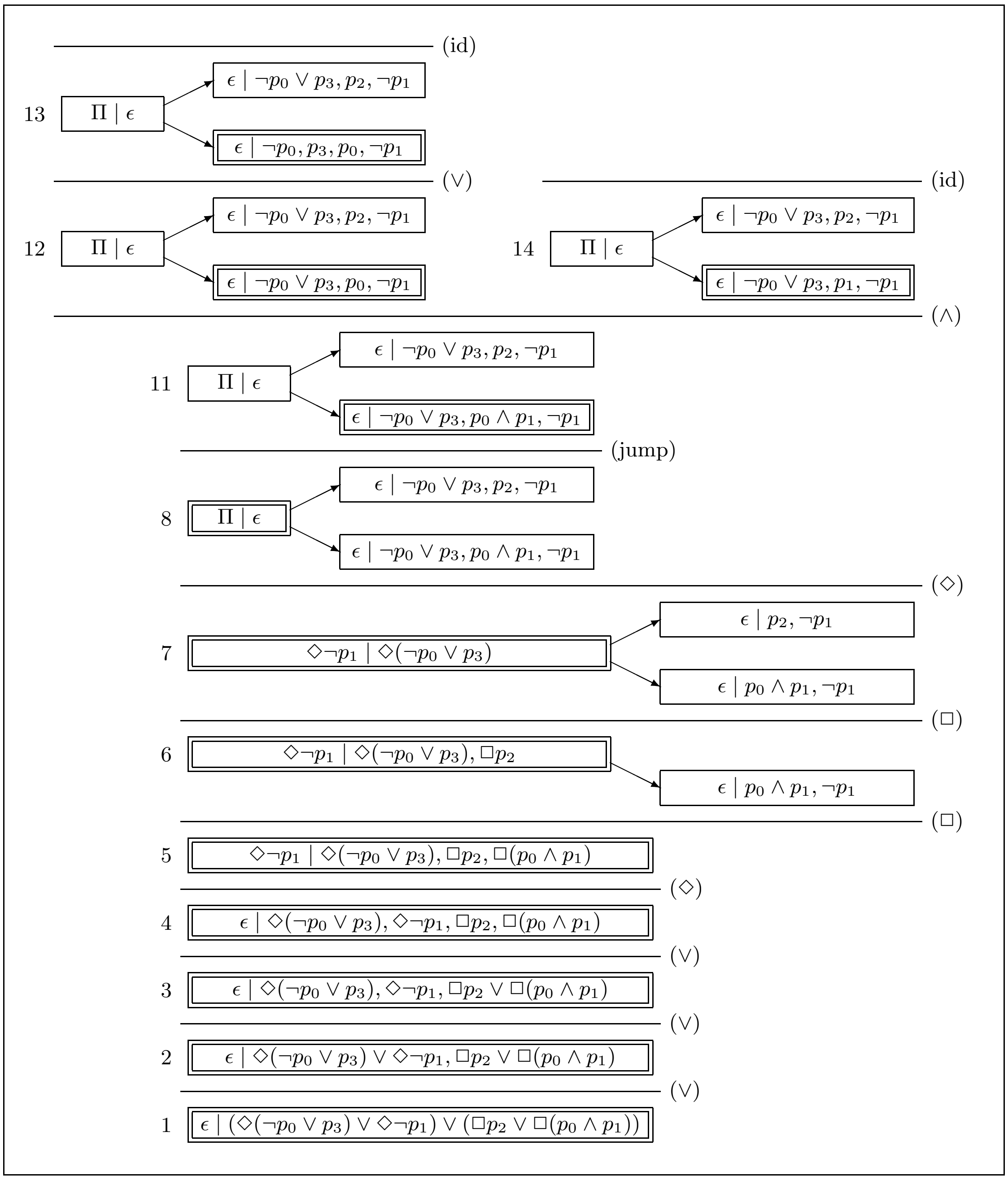
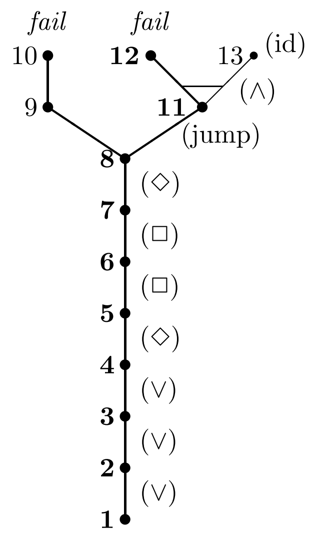
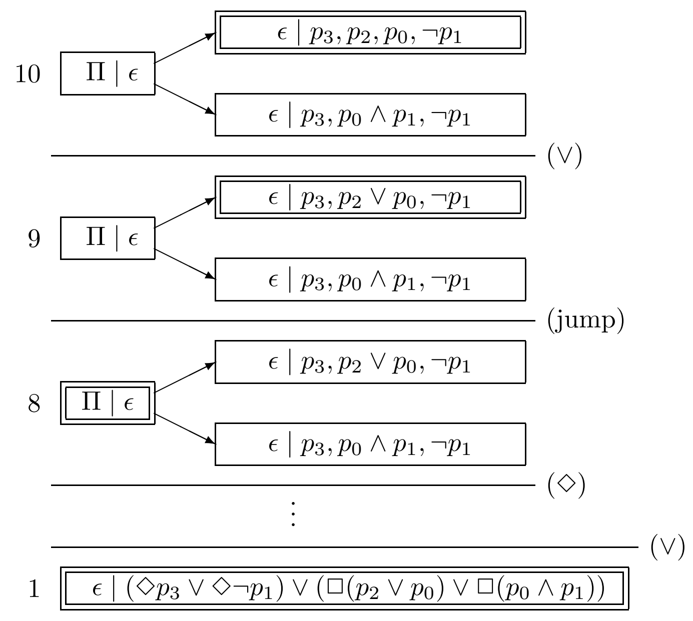

(id)
13 $\Pi \mid \epsilon$
$\epsilon \mid \neg p_0 \vee p_3, p_2, \neg p_1$
$\epsilon \mid \neg p_0, p_3, p_0, \neg p_1$
$(\vee)$
(id)
12 $\Pi \mid \epsilon$
$\epsilon \mid \neg p_0 \vee p_3, p_2, \neg p_1$
$\epsilon \mid \neg p_0 \vee p_3, p_0, \neg p_1$
14 $\Pi \mid \epsilon$
$\epsilon \mid \neg p_0 \vee p_3, p_2, \neg p_1$
$\epsilon \mid \neg p_0 \vee p_3, p_1, \neg p_1$
$(\wedge)$
11 $\Pi \mid \epsilon$
$\epsilon \mid \neg p_0 \vee p_3, p_2, \neg p_1$
$\epsilon \mid \neg p_0 \vee p_3, p_0 \wedge p_1, \neg p_1$
(jump)
8 $\Pi \mid \epsilon$
$\epsilon \mid \neg p_0 \vee p_3, p_2, \neg p_1$
$\epsilon \mid \neg p_0 \vee p_3, p_0 \wedge p_1, \neg p_1$
$(\Diamond)$
7 $\Diamond\neg p_1 \mid \Diamond(\neg p_0 \vee p_3)$
$\epsilon \mid p_2, \neg p_1$
$\epsilon \mid p_0 \wedge p_1, \neg p_1$
$(\Box)$
6 $\Diamond\neg p_1 \mid \Diamond(\neg p_0 \vee p_3), \Box p_2$
$\epsilon \mid p_0 \wedge p_1, \neg p_1$
$(\Box)$
5 $\Diamond\neg p_1 \mid \Diamond(\neg p_0 \vee p_3), \Box p_2, \Box(p_0 \wedge p_1)$
$(\Diamond)$
4 $\epsilon \mid \Diamond(\neg p_0 \vee p_3), \Diamond\neg p_1, \Box p_2, \Box(p_0 \wedge p_1)$
$(\vee)$
3 $\epsilon \mid \Diamond(\neg p_0 \vee p_3), \Diamond\neg p_1, \Box p_2 \vee \Box(p_0 \wedge p_1)$
$(\vee)$
2 $\epsilon \mid \Diamond(\neg p_0 \vee p_3) \vee \Diamond\neg p_1, \Box p_2 \vee \Box(p_0 \wedge p_1)$
$(\vee)$
1 $\epsilon \mid (\Diamond(\neg p_0 \vee p_3) \vee \Diamond\neg p_1) \vee (\Box p_2 \vee \Box(p_0 \wedge p_1))$

Figure 4.l: Proof in $\mathsf{K}^{\mathcal{G},2}$ of the first example in remark 4.2.12. See figure 4.j for the corresponding search tree.



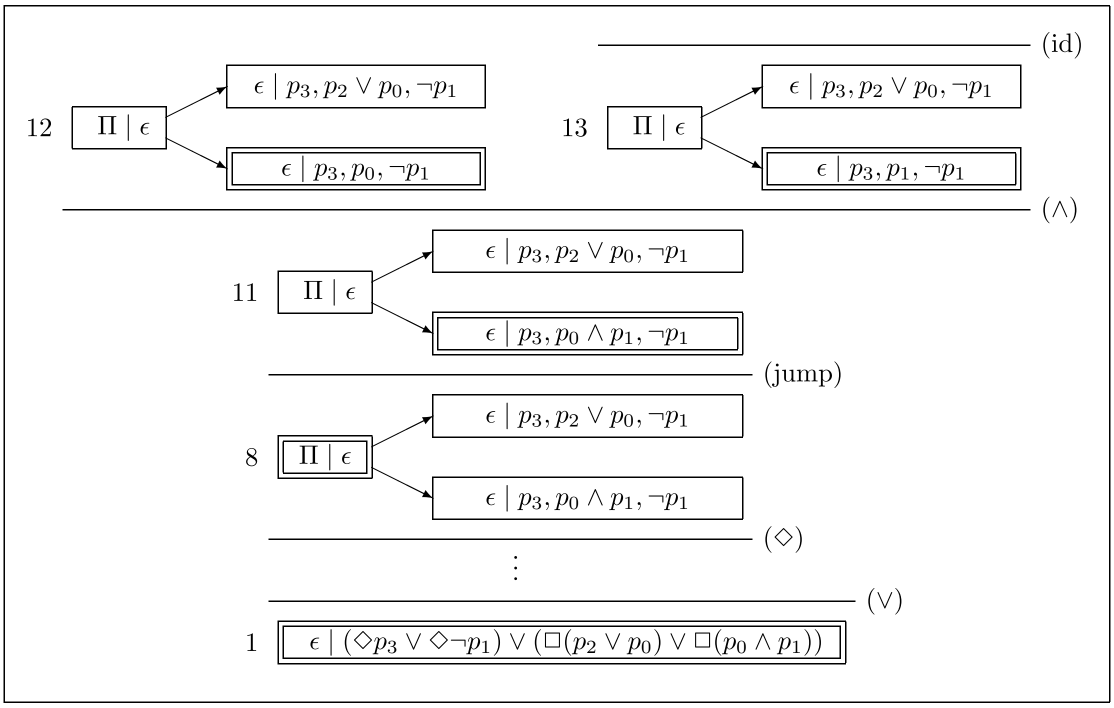
The search starts at node 1 with three (∨) backward applications (node 4). Then two successor worlds are created, and the ◇ formulas are handled (node 8). Now we have to choose between two jumps. One jump ends with a fail (node 9). We backtrack and try the other jump. This branch has another branching because of the (∧) rule and ends in node 11 and 12 . Since only node 12 is an instance of axiom we backtrack again. There remains no other possibility we could try. Thus $\boxed{\boxed{\epsilon \mid A}}$ is not provable in $\mathsf{K}^{\mathcal{G},2}$.
The thick lines are the parts of the search tree that are used for the countermodel.

Figure 4.m: Search tree in $\mathsf{K}^{\mathcal{G},2}$ of the second example in remark 4.2.12. See figure 4.n for the corresponding proof.



10
$\Pi \mid \epsilon$
$\epsilon \mid p_3, p_2, p_0, \neg p_1$
$\epsilon \mid p_3, p_0 \wedge p_1, \neg p_1$
(∨)
9
$\Pi \mid \epsilon$
$\epsilon \mid p_3, p_2 \vee p_0, \neg p_1$
$\epsilon \mid p_3, p_0 \wedge p_1, \neg p_1$
(jump)
8
$\Pi \mid \epsilon$
$\epsilon \mid p_3, p_2 \vee p_0, \neg p_1$
$\epsilon \mid p_3, p_0 \wedge p_1, \neg p_1$
(◇)
(∨)
1
$\epsilon \mid (\Diamond p_3 \vee \Diamond \neg p_1) \vee (\Box(p_2 \vee p_0) \vee \Box(p_0 \wedge p_1))$

Figure 4.n: First non-proof in $\mathsf{K}^{\mathcal{G},2}$ of the second example in remark 4.2.12. See figure 4.m for the corresponding search tree.



(id)
12
$\Pi \mid \epsilon$
$\epsilon \mid p_3, p_2 \vee p_0, \neg p_1$
$\epsilon \mid p_3, p_0, \neg p_1$
13
$\Pi \mid \epsilon$
$\epsilon \mid p_3, p_2 \vee p_0, \neg p_1$
$\epsilon \mid p_3, p_1, \neg p_1$
$(\wedge)$
11
$\Pi \mid \epsilon$
$\epsilon \mid p_3, p_2 \vee p_0, \neg p_1$
$\epsilon \mid p_3, p_0 \wedge p_1, \neg p_1$
(jump)
8
$\Pi \mid \epsilon$
$\epsilon \mid p_3, p_2 \vee p_0, \neg p_1$
$\epsilon \mid p_3, p_0 \wedge p_1, \neg p_1$
$(\Diamond)$
$\vdots$
$(\vee)$
1
$\epsilon \mid (\Diamond p_3 \vee \Diamond \neg p_1) \vee (\Box(p_2 \vee p_0) \vee \Box(p_0 \wedge p_1))$

Figure 4.o: Second non-proof in $\mathsf{K}^{\mathcal{G},2}$ of the second example in remark 4.2.12. See figure 4.j for the corresponding search tree.

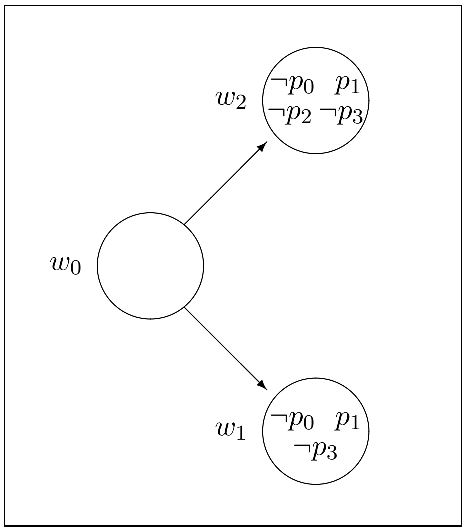
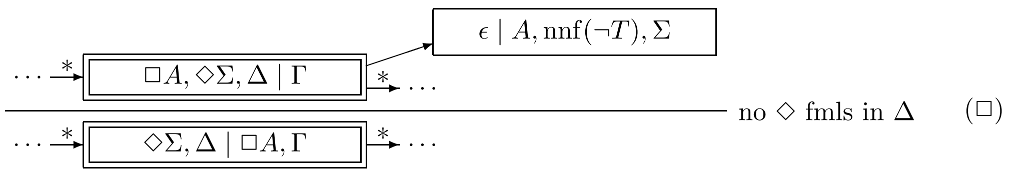


Figure 4.p: Countermodel of the second example in remark 4.2.12. See the figures 4.n and 4.o for the two corresponding non-proofs.


$\epsilon \mid A, \mathrm{nnf}(\neg T), \Sigma$
$\Box A, \Diamond\Sigma, \Delta \mid \Gamma$
$\Diamond\Sigma, \Delta \mid \Box A, \Gamma$
no $\Diamond$ fmls in Δ
$(\Box)$

4.3.3 THEOREM $(\mathsf{K}+T)^{\mathcal{G}}$**: invertible rules**

All rules of $(\mathsf{K}+T)^{\mathcal{G}}$ are invertible.

Proof

Analogous to the proof for $\mathsf{K}^{\mathcal{G}}$ in theorem 4.2.2. The different $(\Box)$ rule makes no difference.

4.3.4 THEOREM equivalence of provability and validity

If A is in negation normal form, then:

$$(\mathsf{K}+T)^{\mathcal{G}} \vdash \boxed{\boxed{\epsilon \mid A, \mathrm{nnf}(\neg T)}} \quad \Leftrightarrow \quad \mathsf{K}+T \models A$$

Proof

'$\Leftarrow$':

We assume that $(\mathsf{K}+T)^{\mathcal{G}} \vdash \boxed{\boxed{\epsilon \mid A, \mathrm{nnf}(\neg T)}}$. As in the case of $\mathsf{K}^{\mathcal{G}}$ we can only obtain trees when doing backward proof search, but proof search in $(\mathsf{K}+T)^{\mathcal{G}}$ does in general not terminate. The restriction of the (jump) and (jump$-$) rules we used for $\mathsf{K}^{\mathcal{G}}$ is not sufficient.

We restrict backward proof search as follows:

- (jump) is only applied backwards if no other rule is applicable.
- (jump) is not applied backwards if the current world already occurs on the branch between the root of the K graph and the current world (loop-check).

Assume for example that during backward proof search we have obtained a K of the form

$$\begin{tikzcd}
\boxed{\Delta \mid \Gamma} \arrow[r, "*"] \arrow[dr] & \cdots \arrow[r] & \boxed{\boxed{\Delta \mid \Gamma}} \\
& \cdots &
\end{tikzcd}$$

Now we are not allowed to apply (jump) backwards because of the second restriction.

The two restrictions make sure that backward proof search always terminates. In order to prove this, we use the function subfmls$'$ defined in the proof of theorem 4.2.4. Let G be a tree we have obtained during backward proof search. If $\boxed{\Delta \mid \Gamma}$ is a vertex of G, then $\mathrm{subfmls}'(\Delta, \Gamma) \subseteq \mathrm{subfmls}'(A, \mathrm{nnf}(\neg T))$. Hence there are only finitely many different vertices, and thus the restriction makes infinite branches impossible.

We take the K graph G at the end of a branch that does not end in an axiom. If in all vertices $\boxed{\Delta \mid \Gamma}$ of G there are only variables and negated variables in Γ, then we can construct a countermodel as in the proof for $\mathsf{K}^{\mathcal{G}}$ in theorem 4.2.5.

Otherwise there are branches of the form

$$\begin{tikzcd}
\boxed{\Delta_1 \mid \Gamma_1} \arrow[r] & \cdots \arrow[r] & \boxed{\Delta_n \mid \Gamma_n} \arrow[r] & \boxed{\Delta \mid \Gamma}
\end{tikzcd}$$

where Γ contains a formula which is neither a variable nor a negated variable. This is only possible if the second restriction was applicable on this branch. Thus $\Gamma = \Gamma_m$ and $\Delta = \Delta_m$ for some $1 \leq m \leq n$. We transform these branches into branches with a loop:

$$\begin{tikzcd}
\boxed{\Delta_1 \mid \Gamma_1} \arrow[r] & \cdots \arrow[r] & \boxed{\Delta_m \mid \Gamma_m} \arrow[r] & \cdots \arrow[r] & \boxed{\Delta_n \mid \Gamma_n} \arrow[ll, bend right]
\end{tikzcd}$$

We convert the resulting K graph into a K model $\langle W, \mathcal{R}, v\rangle$ as in the proof for $\mathsf{K}^{\mathcal{G}}$ in theorem 4.2.5 and prove in the same way that $v(w_0, A) = 0$ for the 'root' w_0 of this model. The fact that $\langle W, \mathcal{R}, v\rangle$ is not a tree does not matter.

It remains to show that the constructed K model is also a $\mathsf{K}+T$ model, i.e. that $\forall w \in W : \forall C \in T : v(w, \mathrm{nnf}(\neg C)) = 0$. Let C be an element of the theory T. When proving $v(w_0, A) = 0$, we proved that if $\boxed{\Delta \mid \Gamma}$ is a vertex in G and $B \in \Gamma, \Delta$, then $v(w(\Delta \mid \Gamma), B) = 0$. Since we add $\mathrm{nnf}(\neg T)$ to each vertex generated by $(\Box)$ and also to the root (see the formulation of this theorem), we know that $\forall w \in W : v(w, \mathrm{nnf}(\neg C)) = 0$.

'$\Rightarrow$':

We can proceed as in the proof for $\mathsf{K}^{\mathcal{G}}$ in theorem 4.2.5. Let $\mathcal{P}$ be a proof of $\boxed{\boxed{\epsilon \mid A}}$ in $\mathsf{K}^{\mathcal{G}}$. (Note that all K graphs that can occur in $\mathcal{P}$ are trees.) We use the same function f as in the proof of theorem 4.2.5 to compute a condition from a K graph.

If G is an instance of an axiom of $(\mathsf{K}+T)^{\mathcal{G}}$, then the condition $f(G)$ is contradictory. If $\frac{G_1}{G_3}$ is an instance of one of the rules $(\vee)$, $(\Diamond)$ and $f(G_1)$ is contradictory, then $f(G_3)$ is contradictory too. If $\frac{G_1 \quad G_2}{G_3}$ is an instance of the rule $(\wedge)$ and both $f(G_1)$ and $f(G_2)$ are contradictory, then $f(G_3)$ is contradictory, too. Applying (jump) or (jump$-$) does not change the condition. So far nothing has changed compared to the corresponding proof for $\mathsf{K}^{\mathcal{G}}$.

The $(\Box)$ rule is the only rule of $(\mathsf{K}+T)^{\mathcal{G}}$ that is different from the corresponding rule of $\mathsf{K}^{\mathcal{G}}$. We assume that $\frac{G_1}{G_3}$ is an instance of the $(\Box)$ rule and that $f(G_1)$ is contradictory. Thus G_1 is of the form

$$\begin{tikzcd}
 & & \boxed{\epsilon \mid A, \mathrm{nnf}(\neg T), \Sigma} \\
\cdots \arrow[r, "*"] & \boxed{\boxed{\Box A, \Diamond\Sigma, \Delta \mid \Gamma}} \arrow[ur] \arrow[r, "*"] & \cdots
\end{tikzcd}$$

and hence G_3 is

$$\begin{tikzcd}
\cdots \arrow[r, "*"] & \boxed{\boxed{\Diamond\Sigma, \Delta \mid \Box A, \Gamma}} \arrow[r, "*"] & \cdots
\end{tikzcd}$$

Thus $f(G_1)$ is of the form

$$\begin{array}{l}
\ldots \\
\text{and } \forall B \in \Box A, \Diamond\Sigma, \Delta, \Gamma : v(w_{\langle l_1,\ldots,l_m\rangle}, B) = 0 \\
\text{and } \ldots \\
\text{and } \exists w_{\langle l_1,\ldots,l_m,k\rangle} \in W : (w_{\langle l_1,\ldots,l_m\rangle} \mathcal{R} w_{\langle l_1,\ldots,l_m,k\rangle} \\
\qquad\qquad\qquad\qquad \text{and } \forall B \in A, \mathrm{nnf}(\neg T), \Sigma : v(w_{\langle l_1,\ldots,l_m,k\rangle}, B) = 0) \\
\text{and } \ldots
\end{array}$$

and $f(G_3)$ then is

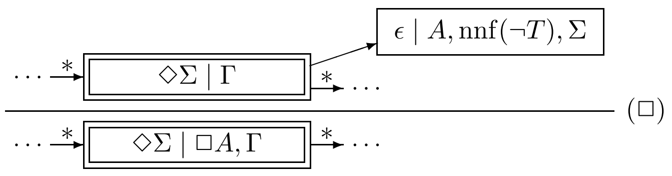
$$\begin{array}{l}
\ldots \\
\text{and } \forall B \in \Box A, \Diamond\Sigma, \Delta, \Gamma : v(w_{\langle l_1,\ldots,l_m\rangle}, B) = 0 \\
\text{and } \ldots
\end{array}$$

The parts abbreviated with '...' are both times the same and concern the vertices that are abbreviated with '...' in G_1 and G_3.

We assume $v(w_{\langle l_1,\dots,l_m\rangle}, \Box A) = 0$ and $v(w_{\langle l_1,\dots,l_m\rangle}, \Diamond\Sigma) = 0$. This implies that there exists a $w_{\langle l_1,\dots,l_m,k\rangle}$ with $w_{\langle l_1,\dots,l_m\rangle}\mathcal{R}w_{\langle l_1,\dots,l_m,k\rangle}$, $v(w_{\langle l_1,\dots,l_m,k\rangle}, A) = 0$, $v(w_{\langle l_1,\dots,l_m,k\rangle}, \Sigma) = 0$, and $\forall B \in T$: $v(w_{\langle l_1,\dots,l_m,k\rangle}, \mathrm{nnf}(\neg B)) = 0$. Therefore also G_3 is contradictory.

4.3.5 DEFINITION graph calculus $(\mathsf{K} + T)^{\mathcal{G},2}$

$(\mathsf{K} + T)^{\mathcal{G},2}$ is the calculus $\mathsf{K}^{\mathcal{G},2}$ with the $(\Box)$ rule replaced by the rule


$\epsilon \mid A, \mathrm{nnf}(\neg T), \Sigma$
$\dots \xrightarrow{*}$
$\Diamond\Sigma \mid \Gamma$
$\xrightarrow{*} \dots$
$(\Box)$
$\dots \xrightarrow{*}$
$\Diamond\Sigma \mid \Box A, \Gamma$
$\xrightarrow{*} \dots$

4.3.6 THEOREM $(\mathsf{K} + T)^{\mathcal{G},2}$: invertible rules

The rule (jump) of $(\mathsf{K} + T)^{\mathcal{G},2}$ is not invertible. All the other rules are strongly invertible.

Proof

Analogous to the proof for $\mathsf{K}^{\mathcal{G}}$ in theorem 4.2.2.

4.3.7 THEOREM equivalence of $(\mathsf{K} + T)^{\mathcal{G}}$ and $(\mathsf{K} + T)^{\mathcal{G},2}$

$$(\mathsf{K} + T)^{\mathcal{G}} \vdash \boxed{\boxed{\epsilon \mid A, \mathrm{nnf}(\neg T)}} \quad\Leftrightarrow\quad (\mathsf{K} + T)^{\mathcal{G},2} \vdash \boxed{\boxed{\epsilon \mid A, \mathrm{nnf}(\neg T)}}$$

Proof

Analogous to the proof for $\mathsf{K}^{\mathcal{G}}$ and $\mathsf{K}^{\mathcal{G},2}$ in theorem 4.2.11.

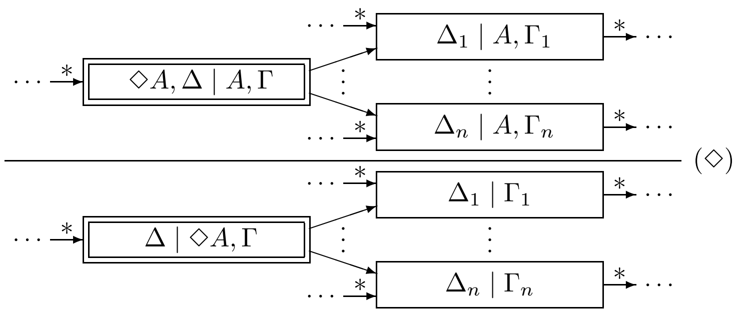
4.4 KT

4.4.1 REMARK from $\mathsf{K}^{\mathcal{G}}$ to $\mathsf{KT}^{\mathcal{G}}$

The only difference between the possible world semantics for K and KT is the reflexivity of the accessibility relation. We only have to change the $(\Diamond)$ rule of $\mathsf{K}^{\mathcal{G}}$ to obtain a graph calculus for $\mathsf{KT}^{\mathcal{G}}$, since now $\Diamond A$ says something about the current world and not just about the 'successor' worlds.

4.4.2 DEFINITION graph calculus $\mathsf{KT}^{\mathcal{G}}$

The calculus $\mathsf{KT}^{\mathcal{G}}$ is the calculus $\mathsf{K}^{\mathcal{G}}$ with the $(\Diamond)$ rule replaced by the following rule:



$$\cfrac{\cfrac{\cfrac{}{\boxed{\Diamond\neg p_0, \Diamond\neg p_0 \vee p_0 \mid \neg p_0, p_0}}\,(\mathrm{id})}{\boxed{\Diamond\neg p_0 \vee p_0 \mid \Diamond\neg p_0, p_0}}\,(\Diamond)}{\boxed{\epsilon \mid \Diamond\neg p_0 \vee p_0}}\,(\vee)$$

Figure 4.q: The proof from the example in definition 4.4.2.

Example

The axiom $\Box A \to A$ is the only difference between the two Hilbert-style calculi $\mathsf{K}^{\mathcal{H}}$ and $\mathsf{KT}^{\mathcal{H}}$. The proof in figure 4.q shows that $\mathsf{KT}^{\mathcal{G}} \vdash \boxed{\epsilon \mid \mathrm{nnf}(\Box p_0 \to p_0)}$. It is not a proof in $\mathsf{K}^{\mathcal{G}}$ because of the different $(\Diamond)$ rule, which does not put $\neg p_0$ into the current world.

4.4.3 THEOREM $\mathsf{KT}^{\mathcal{G}}$: invertible rules

All rules of $\mathsf{KT}^{\mathcal{G}}$ are invertible.

Proof

Analogous to the proof for $\mathsf{K}^{\mathcal{G}}$ in theorem 4.2.2. The difference in the $(\Diamond)$ rule causes no problems.

4.4.4 THEOREM $\mathsf{KT}^{\mathcal{G}}$: termination

In general, backward proof search in $\mathsf{KT}^{\mathcal{G}}$ does not terminate since the rules (jump) and (jump−) are always applicable.

However, backward proof search in $\mathsf{KT}^{\mathcal{G}}$ terminates if we add the following two restrictions:

- (jump) is only applied backwards if, perhaps after further backward applications of (jump), a rule different from (jump) and (jump−) is applicable backwards.
- (jump−) is never applied backwards right after (jump).

These restrictions do not change the set of provable vertices $\boxed{\epsilon \mid A}$. Note that still no backtracking is required.

Proof

We can use the same proof as for theorem 4.2.4.

4.4.5 THEOREM equivalence of provability and validity

If A is in negation normal form, then:

$$\mathsf{KT}^{\mathcal{G}} \vdash \boxed{\epsilon \mid A} \quad \Leftrightarrow \quad \mathsf{KT} \models A$$

Proof

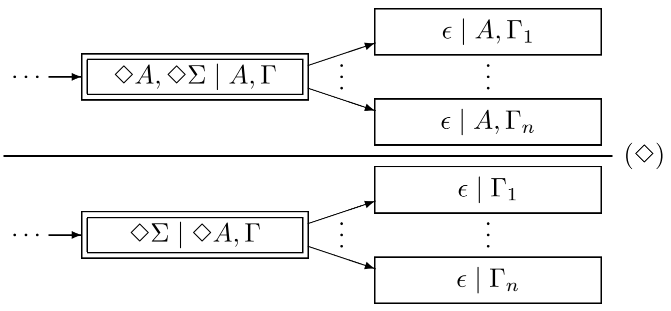
Analogous to the proof for $\mathsf{K}^{\mathcal{G}}$ in theorem 4.2.5. The different $(\Diamond)$ rule corresponds to the reflexivity of the accessibility relation.

4.4.6 REMARK from $\mathsf{KT}^{\mathcal{G}}$ to $\mathsf{KT}^{\mathcal{G},2}$

It is possible to simplify the calculus $\mathsf{KT}^{\mathcal{G}}$ in the same way that we simplified the calculus $\mathsf{K}^{\mathcal{G}}$.

4.4.7 DEFINITION graph calculus $\mathsf{KT}^{\mathcal{G},2}$

The calculus $\mathsf{KT}^{\mathcal{G},2}$ is the calculus $\mathsf{K}^{\mathcal{G},2}$ with the ($\Diamond$) rule replaced by the following rule:



$\cdots$
$\Diamond A, \Diamond\Sigma \mid A, \Gamma$
$\epsilon \mid A, \Gamma_1$
$\epsilon \mid A, \Gamma_n$
($\Diamond$)
$\cdots$
$\Diamond\Sigma \mid \Diamond A, \Gamma$
$\epsilon \mid \Gamma_1$
$\epsilon \mid \Gamma_n$

4.4.8 THEOREM $\mathsf{KT}^{\mathcal{G},2}$: invertible rules

The rule (jump) of $\mathsf{KT}^{\mathcal{G},2}$ is not invertible. All the other rules are strongly invertible.

Proof

Analogous to the proof for $\mathsf{K}^{\mathcal{G}}$ in theorem 4.2.10.

4.4.9 THEOREM equivalence of $\mathsf{KT}^{\mathcal{G}}$ and $\mathsf{KT}^{\mathcal{G},2}$

$$\mathsf{KT}^{\mathcal{G}} \vdash \boxed{\boxed{\epsilon \mid A}} \quad \Leftrightarrow \quad \mathsf{KT}^{\mathcal{G},2} \vdash \boxed{\boxed{\epsilon \mid A}}$$

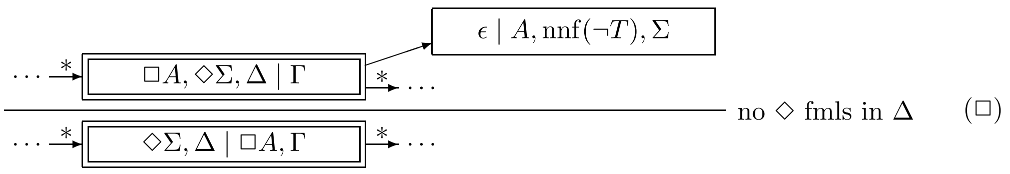
Proof

Analogous to the proof for $\mathsf{K}^{\mathcal{G}}$ and $\mathsf{K}^{\mathcal{G},2}$ in theorem 4.2.11.

4.5 $\mathsf{KT} + T$

4.5.1 DEFINITION graph calculus $(\mathsf{KT} + T)^{\mathcal{G}}$

$(\mathsf{KT} + T)^{\mathcal{G}}$ is the calculus $\mathsf{KT}^{\mathcal{G}}$ with the ($\Box$) rule replaced by the rule



$\cdots \xrightarrow{*}$
$\Box A, \Diamond\Sigma, \Delta \mid \Gamma$
$\xrightarrow{*} \cdots$
$\epsilon \mid A, \mathrm{nnf}(\neg T), \Sigma$
no $\Diamond$ fmls in Δ
($\Box$)
$\cdots \xrightarrow{*}$
$\Diamond\Sigma, \Delta \mid \Box A, \Gamma$
$\xrightarrow{*} \cdots$

4.5.2 THEOREM $(\mathsf{KT} + T)^{\mathcal{G}}$: invertible rules

All rules of $(\mathsf{KT} + T)^{\mathcal{G}}$ are invertible.

Proof

Analogous to the proof for $\mathsf{K}^{\mathcal{G}}$ in theorem 4.2.2.

4.5.3 THEOREM equivalence of provability and validity

If A is in negation normal form, then:

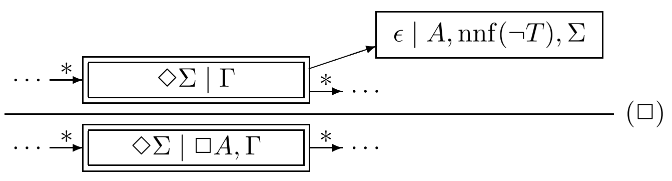
$$(\mathsf{KT}+T)^{\mathcal{G}} \vdash \boxed{\epsilon \mid A, \mathrm{nnf}(\neg T)} \quad\Leftrightarrow\quad \mathsf{KT}+T \models A$$

Proof

Analogous to the proof for $(\mathsf{K}+T)^{\mathcal{G}}$ in theorem 4.3.4.

4.5.4 DEFINITION graph calculus $(\mathsf{KT}+T)^{\mathcal{G},2}$

$(\mathsf{KT}+T)^{\mathcal{G},2}$ is the calculus $\mathsf{KT}^{\mathcal{G},2}$ with the $(\Box)$ rule replaced by the rule



$\epsilon \mid A, \mathrm{nnf}(\neg T), \Sigma$
$\cdots \xrightarrow{*}$
$\Diamond\Sigma \mid \Gamma$
$\xrightarrow{*} \cdots$
$(\Box)$
$\cdots \xrightarrow{*}$
$\Diamond\Sigma \mid \Box A, \Gamma$
$\xrightarrow{*} \cdots$

4.5.5 THEOREM $(\mathsf{KT}+T)^{\mathcal{G},2}$: invertible rules

The rule (jump) of $(\mathsf{KT}+T)^{\mathcal{G},2}$ is not invertible. All the other rules are strongly invertible.

Proof

Analogous to the proof for $\mathsf{KT}^{\mathcal{G}}$ in theorem 4.2.2.

4.5.6 THEOREM equivalence of $(\mathsf{KT}+T)^{\mathcal{G}}$ and $(\mathsf{KT}+T)^{\mathcal{G},2}$

$$(\mathsf{KT}+T)^{\mathcal{G}} \vdash \boxed{\epsilon \mid A, \mathrm{nnf}(\neg T)} \quad\Leftrightarrow\quad (\mathsf{KT}+T)^{\mathcal{G},2} \vdash \boxed{\epsilon \mid A, \mathrm{nnf}(\neg T)}$$

Proof

Analogous to the proof for $\mathsf{KT}^{\mathcal{G},2}$ in theorem 4.2.11.

4.6 S4

4.6.1 REMARK from $\mathsf{KT}^{\mathcal{G}}$ to $\mathsf{S4}^{\mathcal{G}}$

The difference between the possible world semantics for KT and S4 is the transitivity of the accessibility relation. Therefore in the graph calculus the $\Diamond$ is still present in the successor of the current world.

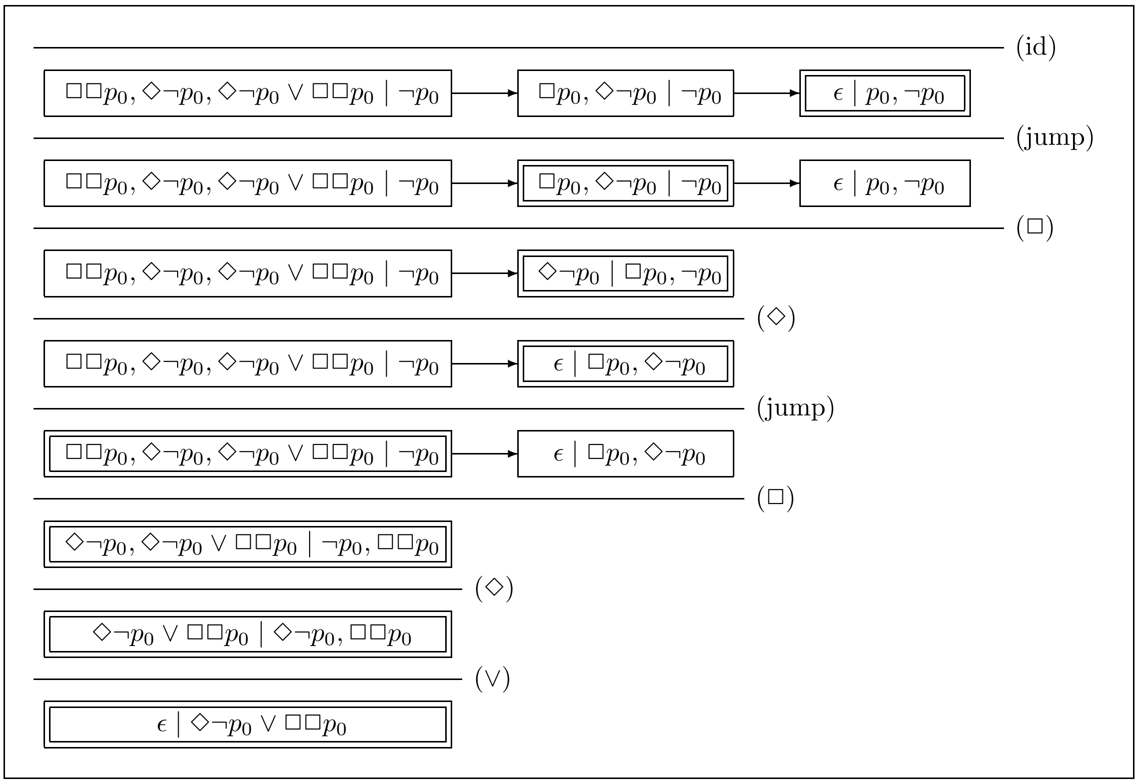
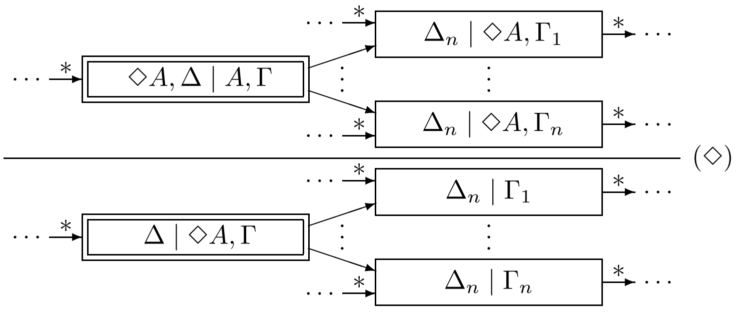

(id)
□□p₀, ◇¬p₀, ◇¬p₀ ∨ □□p₀ | ¬p₀
□p₀, ◇¬p₀ | ¬p₀
ϵ | p₀, ¬p₀
(jump)
□□p₀, ◇¬p₀, ◇¬p₀ ∨ □□p₀ | ¬p₀
□p₀, ◇¬p₀ | ¬p₀
ϵ | p₀, ¬p₀
(□)
□□p₀, ◇¬p₀, ◇¬p₀ ∨ □□p₀ | ¬p₀
◇¬p₀ | □p₀, ¬p₀
(◇)
□□p₀, ◇¬p₀, ◇¬p₀ ∨ □□p₀ | ¬p₀
ϵ | □p₀, ◇¬p₀
(jump)
□□p₀, ◇¬p₀, ◇¬p₀ ∨ □□p₀ | ¬p₀
ϵ | □p₀, ◇¬p₀
(□)
◇¬p₀, ◇¬p₀ ∨ □□p₀ | ¬p₀, □□p₀
(◇)
◇¬p₀ ∨ □□p₀ | ◇¬p₀, □□p₀
(∨)
ϵ | ◇¬p₀ ∨ □□p₀

Figure 4.r: The proof from the example in definition 4.6.2.

4.6.2 DEFINITION graph calculus $\mathsf{S4}^{\mathcal{G}}$

The calculus $\mathsf{S4}^{\mathcal{G}}$ is the calculus $\mathsf{KT}^{\mathcal{G}}$ with the ($\Diamond$) rule replaced by the following rule:



Example

The axiom $\Box A \to \Box\Box A$ is the only difference between the two Hilbert-style calculi $\mathsf{KT}^{\mathcal{H}}$ and $\mathsf{S4}^{\mathcal{H}}$. The proof in figure 4.r shows that $\mathsf{S4}^{\mathcal{G}} \vdash \boxed{\boxed{\epsilon \mid \mathrm{nnf}(\Box p_0 \to \Box\Box p_0)}}$. If we used $\mathsf{KT}^{\mathcal{G}}$ instead, then we would have $\neg p_0$ instead of $\Diamond\neg p_0$ in the second world because of the different ($\Box$) rule, and thus no $\neg p_0$ in the third world.

4.6.3 THEOREM $\mathsf{S4}^{\mathcal{G}}$: invertible rules

All rules of $\mathsf{S4}^{\mathcal{G}}$ are invertible.

Proof

Analogous to the proof for $\mathsf{K}^{\mathcal{G}}$ in theorem 4.2.2. The difference in the ($\Diamond$) rule causes no problems.

4.6.4 THEOREM equivalence of provability and validity

If A is in negation normal form, then:

$$\mathsf{S4}^{\mathcal{G}} \vdash \boxed{\boxed{\epsilon \mid A}} \quad \Leftrightarrow \quad \mathsf{S4} \models A$$

Proof

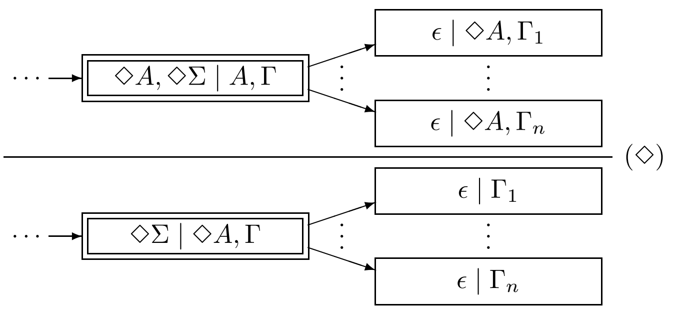
Backward proof search in $\mathsf{S4}^{\mathcal{G}}$ does in general not terminate. Therefore we have to proceed as in the proof for $(\mathsf{K}+T)^{\mathcal{G}}$ in theorem 4.3.4.

4.6.5 REMARK from $\mathsf{S4}^{\mathcal{G}}$ to $\mathsf{S4}^{\mathcal{G},2}$

It is possible to simplify the calculus $\mathsf{S4}^{\mathcal{G}}$ in the same way that we simplified the calculi $\mathsf{K}^{\mathcal{G}}$ and $\mathsf{KT}^{\mathcal{G}}$.

4.6.6 DEFINITION graph calculus $\mathsf{S4}^{\mathcal{G},2}$

The calculus $\mathsf{S4}^{\mathcal{G},2}$ is the calculus $\mathsf{KT}^{\mathcal{G},2}$ with the $(\Diamond)$ rule replaced by the following rule:



$\cdots$
$\Diamond A, \Diamond\Sigma \mid A, \Gamma$
$\epsilon \mid \Diamond A, \Gamma_1$
$\epsilon \mid \Diamond A, \Gamma_n$
$(\Diamond)$
$\cdots$
$\Diamond\Sigma \mid \Diamond A, \Gamma$
$\epsilon \mid \Gamma_1$
$\epsilon \mid \Gamma_n$

4.6.7 THEOREM $\mathsf{S4}^{\mathcal{G},2}$: invertible rules

The rule (jump) is of $\mathsf{S4}^{\mathcal{G},2}$ is not invertible. All the other rules are strongly invertible.

Proof

Analogous to the proof for $\mathsf{K}^{\mathcal{G},2}$in theorem 4.2.10.

4.6.8 THEOREM equivalence of $\mathsf{S4}^{\mathcal{G}}$ and $\mathsf{S4}^{\mathcal{G},2}$

$$\mathsf{S4}^{\mathcal{G}} \vdash \boxed{\boxed{\epsilon \mid A}} \quad \Leftrightarrow \quad \mathsf{S4}^{\mathcal{G},2} \vdash \boxed{\boxed{\epsilon \mid A}}$$

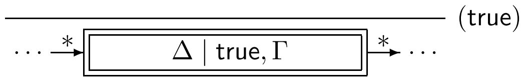
Proof

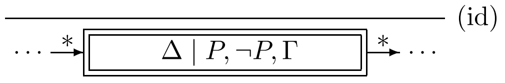
Analogous to the proof for $\mathsf{K}^{\mathcal{G}}$ and $\mathsf{K}^{\mathcal{G},2}$ in theorem 4.2.11.

4.7 K_t

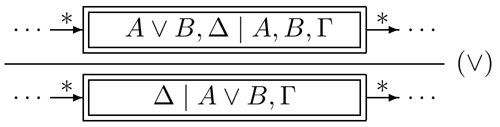
4.7.1 DEFINITION $\mathsf{K}_t^{\mathcal{G}}$

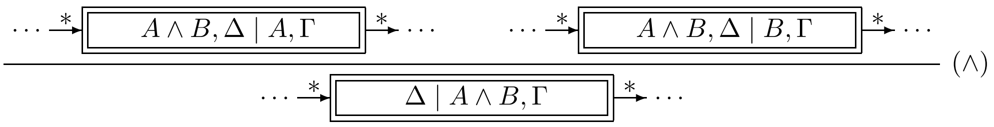
axioms:

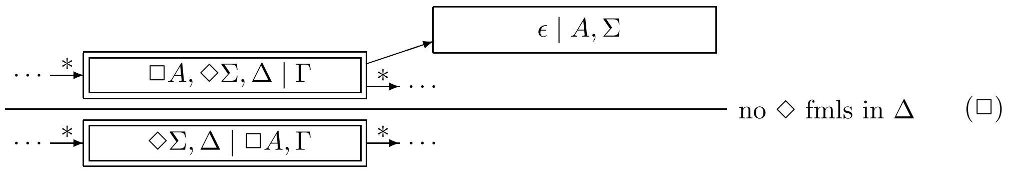


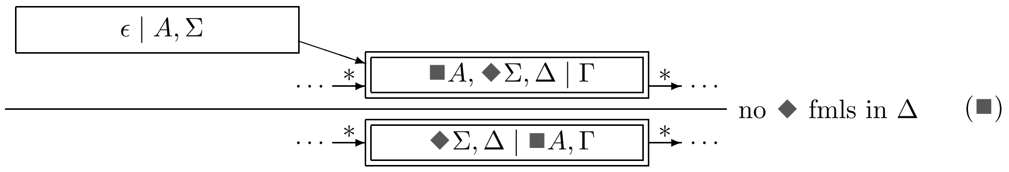


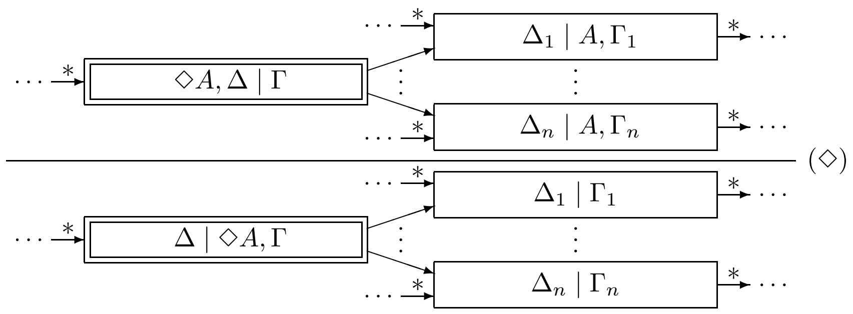
rules:

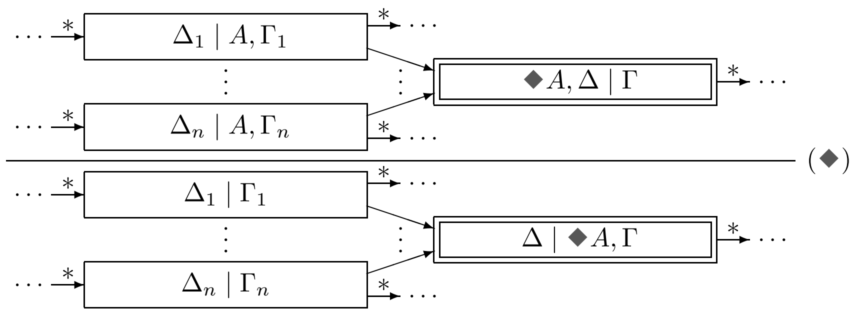

$A \vee B, \Delta \mid A, B, \Gamma$
$\Delta \mid A \vee B, \Gamma$
$(\vee)$

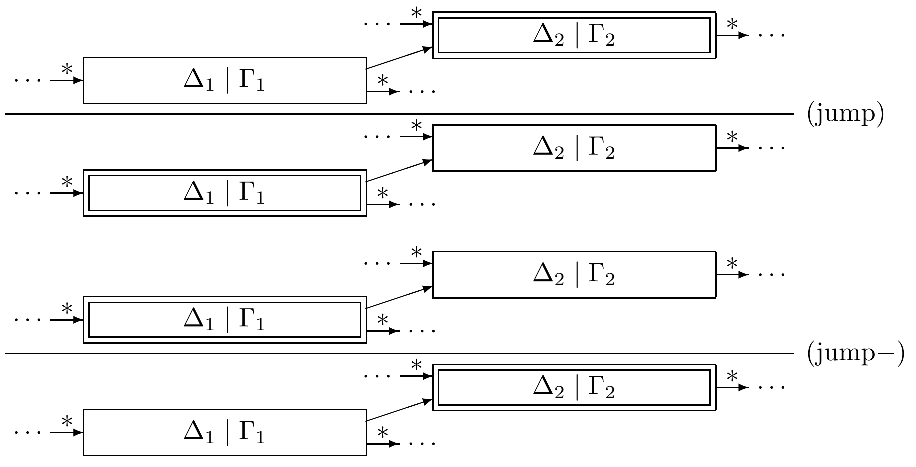

$A \wedge B, \Delta \mid A, \Gamma$
$A \wedge B, \Delta \mid B, \Gamma$
$\Delta \mid A \wedge B, \Gamma$
$(\wedge)$


$\epsilon \mid A, \Sigma$
$\Box A, \Diamond\Sigma, \Delta \mid \Gamma$
$\Diamond\Sigma, \Delta \mid \Box A, \Gamma$
no $\Diamond$ fmls in Δ
$(\Box)$


$\epsilon \mid A, \Sigma$
$\blacksquare A, \blacklozenge\Sigma, \Delta \mid \Gamma$
$\blacklozenge\Sigma, \Delta \mid \blacksquare A, \Gamma$
no $\blacklozenge$ fmls in Δ
$(\blacksquare)$


$\Diamond A, \Delta \mid \Gamma$
$\Delta_1 \mid A, \Gamma_1$
$\Delta_n \mid A, \Gamma_n$
$\Delta \mid \Diamond A, \Gamma$
$\Delta_1 \mid \Gamma_1$
$\Delta_n \mid \Gamma_n$
$(\Diamond)$


$\Delta_1 \mid A, \Gamma_1$
$\Delta_n \mid A, \Gamma_n$
$\blacklozenge A, \Delta \mid \Gamma$
$\Delta_1 \mid \Gamma_1$
$\Delta_n \mid \Gamma_n$
$\Delta \mid \blacklozenge A, \Gamma$
$(\blacklozenge)$



main formulas: true in (true), P and $\neg P$ in (id), $A \vee B$ in ($\vee$), $A \wedge B$ in ($\wedge$), $\Box A$ in ($\Box$), $\blacksquare A$ in ($\blacksquare$), $\Diamond A$ in ($\Diamond$) $\blacklozenge A$ in ($\blacklozenge$), none in (jump) and (jump−)

4.7.2 THEOREM $\mathsf{K}_t^{\mathcal{G}}$: invertible rules

All rules of $\mathsf{K}_t^{\mathcal{G}}$ are invertible.

Proof

Analogous to the proof for $\mathsf{K}^{\mathcal{G}}$ in theorem 4.2.2.

4.7.3 REMARK $\mathsf{K}_t^{\mathcal{G}}$: termination

In general, backward proof search in $\mathsf{K}_t^{\mathcal{G}}$ does not terminate, even if we restrict the usage of the rules (jump) and (jump−) such that ‘useless’ jumps are no longer possible. If we added the condition $\Box A \notin \Sigma$ resp. $\blacksquare A \notin \Sigma$ to the rules ($\Box$) resp. ($\blacksquare$), then the backward proof search would always terminate. This has been observed in [Zim94].

Example

Backward proof search for $\boxed{\boxed{\epsilon \mid \Box p_0 \vee \Diamond\blacklozenge\Box p_0}}$ in $\mathsf{K}_t^{\mathcal{G}}$ does not terminate (see figure 4.s).
If we put the $\Box$ formulas on the left hand side when applying ($\Box$) backwards and add the condition $\Box A \notin \Sigma$ to the rule ($\Box$) of $\mathsf{K}_t^{\mathcal{G}}$, then the search terminates (see figure 4.t). Both times we use the abbreviation $B \equiv \Box p_0 \vee \Diamond\blacklozenge\Box p_0$.

4.7.4 THEOREM equivalence of provability and validity

If A is in negation normal form, then:

$$\mathsf{K}_t^{\mathcal{G}} \vdash \boxed{\boxed{\epsilon \mid A}} \quad \Leftrightarrow \quad \mathsf{K}_t \models A$$

Proof

Analogous to the proof for $\mathsf{K}^{\mathcal{G}}$ in theorem 4.2.5. The fact that backward proof search in $\mathsf{K}_t^{\mathcal{G}}$ does in general not terminate causes no problem, since we can easily restrict the calculus to ensure termination while still obtaining a countermodel in the case of a failure. See remark 4.7.3 for details.

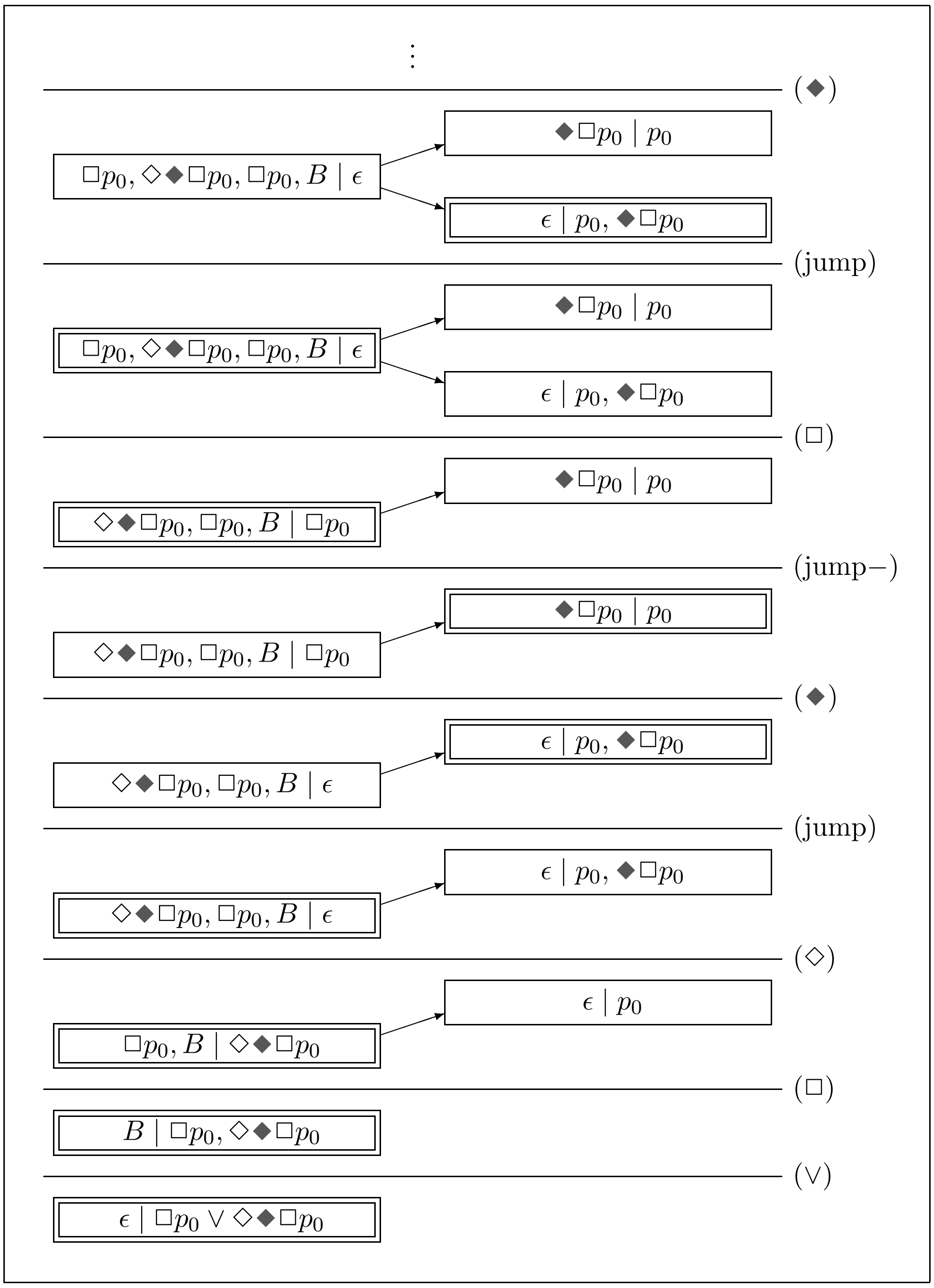

$\vdots$
$(\blacklozenge)$
$\blacklozenge\Box p_0 \mid p_0$
$\Box p_0, \Diamond\blacklozenge\Box p_0, \Box p_0, B \mid \epsilon$
$\epsilon \mid p_0, \blacklozenge\Box p_0$
(jump)
$\blacklozenge\Box p_0 \mid p_0$
$\Box p_0, \Diamond\blacklozenge\Box p_0, \Box p_0, B \mid \epsilon$
$\epsilon \mid p_0, \blacklozenge\Box p_0$
$(\Box)$
$\blacklozenge\Box p_0 \mid p_0$
$\Diamond\blacklozenge\Box p_0, \Box p_0, B \mid \Box p_0$
(jump−)
$\blacklozenge\Box p_0 \mid p_0$
$\Diamond\blacklozenge\Box p_0, \Box p_0, B \mid \Box p_0$
$(\blacklozenge)$
$\epsilon \mid p_0, \blacklozenge\Box p_0$
$\Diamond\blacklozenge\Box p_0, \Box p_0, B \mid \epsilon$
(jump)
$\epsilon \mid p_0, \blacklozenge\Box p_0$
$\Diamond\blacklozenge\Box p_0, \Box p_0, B \mid \epsilon$
$(\Diamond)$
$\epsilon \mid p_0$
$\Box p_0, B \mid \Diamond\blacklozenge\Box p_0$
$(\Box)$
$B \mid \Box p_0, \Diamond\blacklozenge\Box p_0$
$(\vee)$
$\epsilon \mid \Box p_0 \vee \Diamond\blacklozenge\Box p_0$

Figure 4.s: The non-terminating backward proof search in $\mathsf{K}_t^{\mathcal{G}}$ of the example in remark 4.7.3.

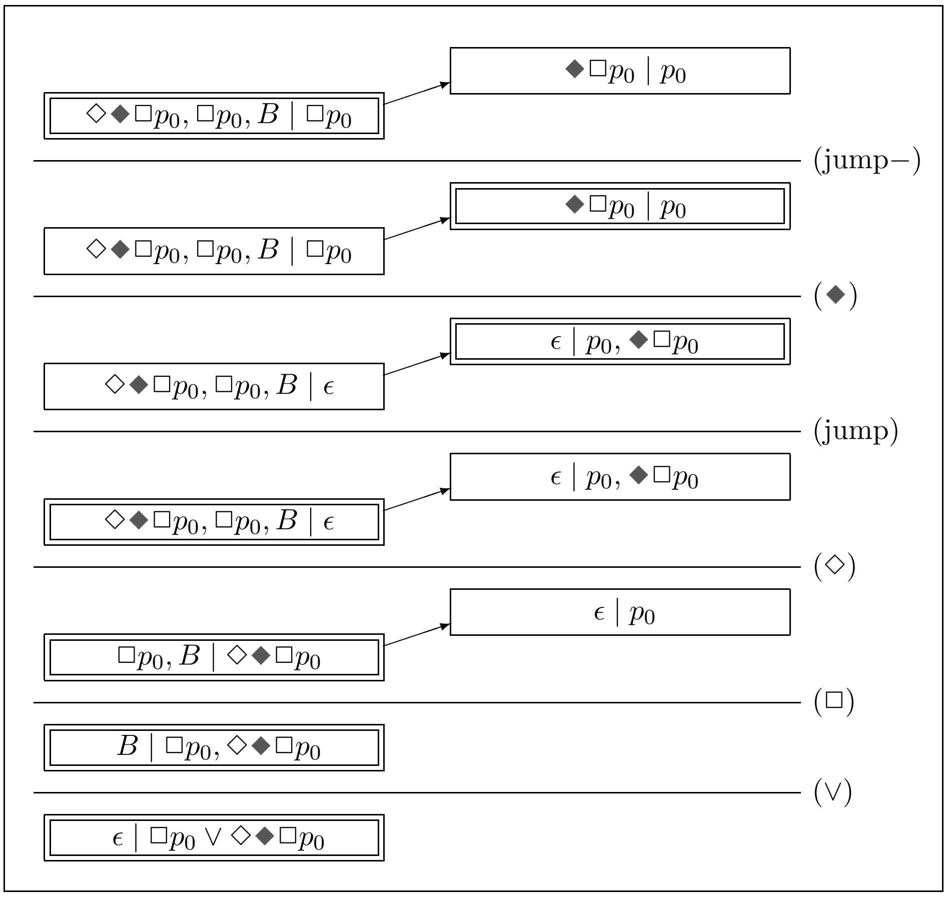
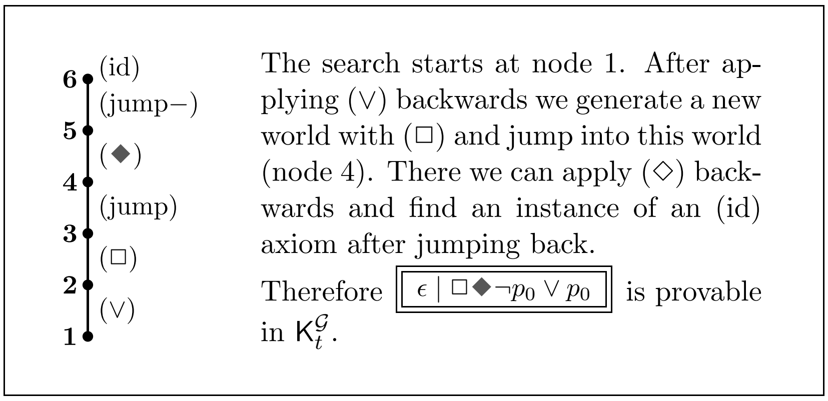

$\blacklozenge\Box p_0 \mid p_0$
$\Diamond\blacklozenge\Box p_0, \Box p_0, B \mid \Box p_0$
(jump$-$)
$\blacklozenge\Box p_0 \mid p_0$
$\Diamond\blacklozenge\Box p_0, \Box p_0, B \mid \Box p_0$
($\blacklozenge$)
$\epsilon \mid p_0, \blacklozenge\Box p_0$
$\Diamond\blacklozenge\Box p_0, \Box p_0, B \mid \epsilon$
(jump)
$\epsilon \mid p_0, \blacklozenge\Box p_0$
$\Diamond\blacklozenge\Box p_0, \Box p_0, B \mid \epsilon$
($\Diamond$)
$\epsilon \mid p_0$
$\Box p_0, B \mid \Diamond\blacklozenge\Box p_0$
($\Box$)
$B \mid \Box p_0, \Diamond\blacklozenge\Box p_0$
($\vee$)
$\epsilon \mid \Box p_0 \vee \Diamond\blacklozenge\Box p_0$

Figure 4.t: The terminating backward proof search in a modified version of $\mathsf{K}_t^{\mathcal{G}}$ of the example in remark 4.7.3.



The search starts at node 1. After applying (∨) backwards we generate a new world with (□) and jump into this world (node 4). There we can apply (◇) backwards and find an instance of an (id) axiom after jumping back.

Therefore $\boxed{\boxed{\epsilon \mid \Box\blacklozenge\neg p_0 \lor p_0}}$ is provable in $\mathsf{K}_t^{\mathcal{G}}$.

Figure 4.u: The search tree in $\mathsf{K}_t^{\mathcal{G}}$ of the first example in remark 4.7.5. See figure 4.v for the corresponding proof.

4.7.5 REMARK $\mathsf{K}_t^{\mathcal{G}}$: backward proof search

Except for the non-termination problem discussed in remark 4.7.3, backward proof search in $\mathsf{K}_t^{\mathcal{G}}$ is similar to backward proof search in $\mathsf{K}^{\mathcal{G}}$. Since all rules are invertible, we know that if one branch fails, then the formula is not provable.

Example

First we do proof search for $\boxed{\boxed{\epsilon \mid \Box\blacklozenge\neg p_0 \lor p_0}}$. The formula $\Box\blacklozenge\neg p_0 \lor p_0$ is the negation normal form of the formula $\Diamond\blacksquare p_0 \to p_0$. See figure 4.u for the search tree and figure 4.v for the corresponding proof. We use the abbreviation $A \equiv \Box\blacklozenge\neg p_0 \lor p_0$.

Now we check whether $\boxed{\boxed{\epsilon \mid \Box(\blacklozenge\neg p_0 \land \blacksquare\Diamond\neg p_1) \lor p_0 \lor p_1}}$ is provable in $\mathsf{K}^{\mathcal{G}}$. The search fails. We obtain the search tree in figure 4.w and the non-proof in figure 4.x. We abbreviate the formulas on the left hand side of the vertices by '. . . ' if they are neither $\Diamond$ nor $\blacklozenge$ formulas.

To obtain a countermodel $\mathcal{M}$, we take the K_t graph in node 12 in figure 4.w, remove the left hand side in each rule and negate the literals (see figure 4.y). Indeed $\mathcal{M}, w_1 \models \blacklozenge\Box p_1$, thus $\mathcal{M}, w_0 \models \Diamond(\blacksquare p_0 \lor \blacklozenge\Box p_1) \land \neg p_0 \land \neg p_1$, i.e. $\mathcal{M}, w_0 \models \neg(\Box(\blacklozenge\neg p_0 \land \blacksquare\Diamond\neg p_1) \lor p_0 \lor p_1)$.

4.7.6 REMARK simplifying $\mathsf{K}_t^{\mathcal{G}}$

It is not possible to simplify $\mathsf{K}_t^{\mathcal{G}}$ in the same way as $\mathsf{K}^{\mathcal{G}}$, as we can eliminate neither (jump−) nor (jump). Therefore we cannot omit arrows as we could for example in the (∨) rule of $\mathsf{K}^{\mathcal{G},2}$.

Example

In the proof of $\boxed{\boxed{\epsilon \mid \Box\blacklozenge\neg p_0 \lor p_0}}$ in $\mathsf{K}_t^{\mathcal{G},2}$ in the first example of remark 4.7.5 we need both (jump) and (jump−).

4.8 OTHER LOGICS

4.8.1 REMARK S5

If we develop a graph calculus for $\mathsf{S5}$, then we obtain a notational variant of the hypersequent calculus. A similar check as for K_t is needed to ensure termination.

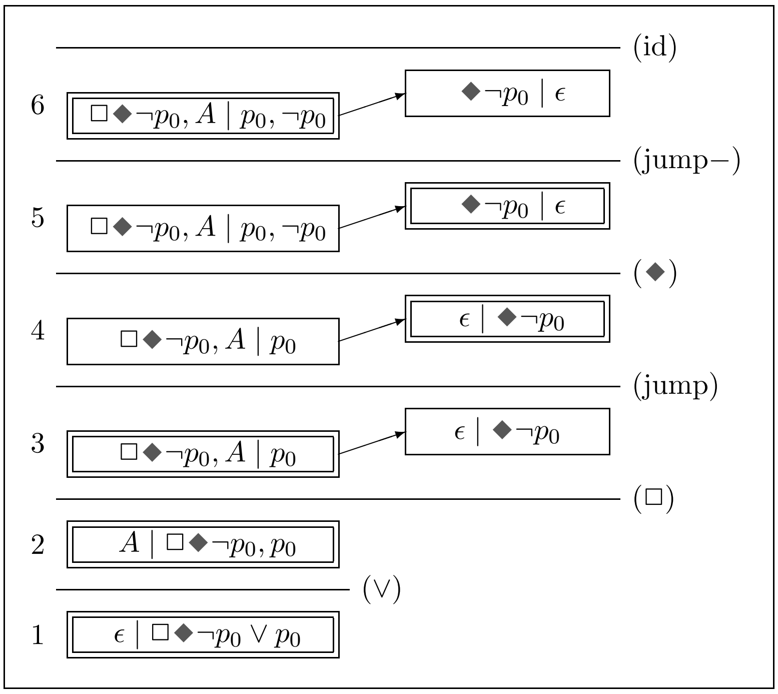
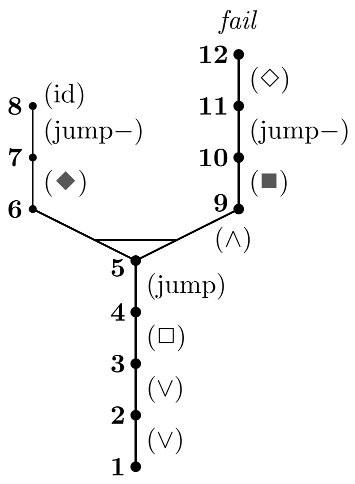
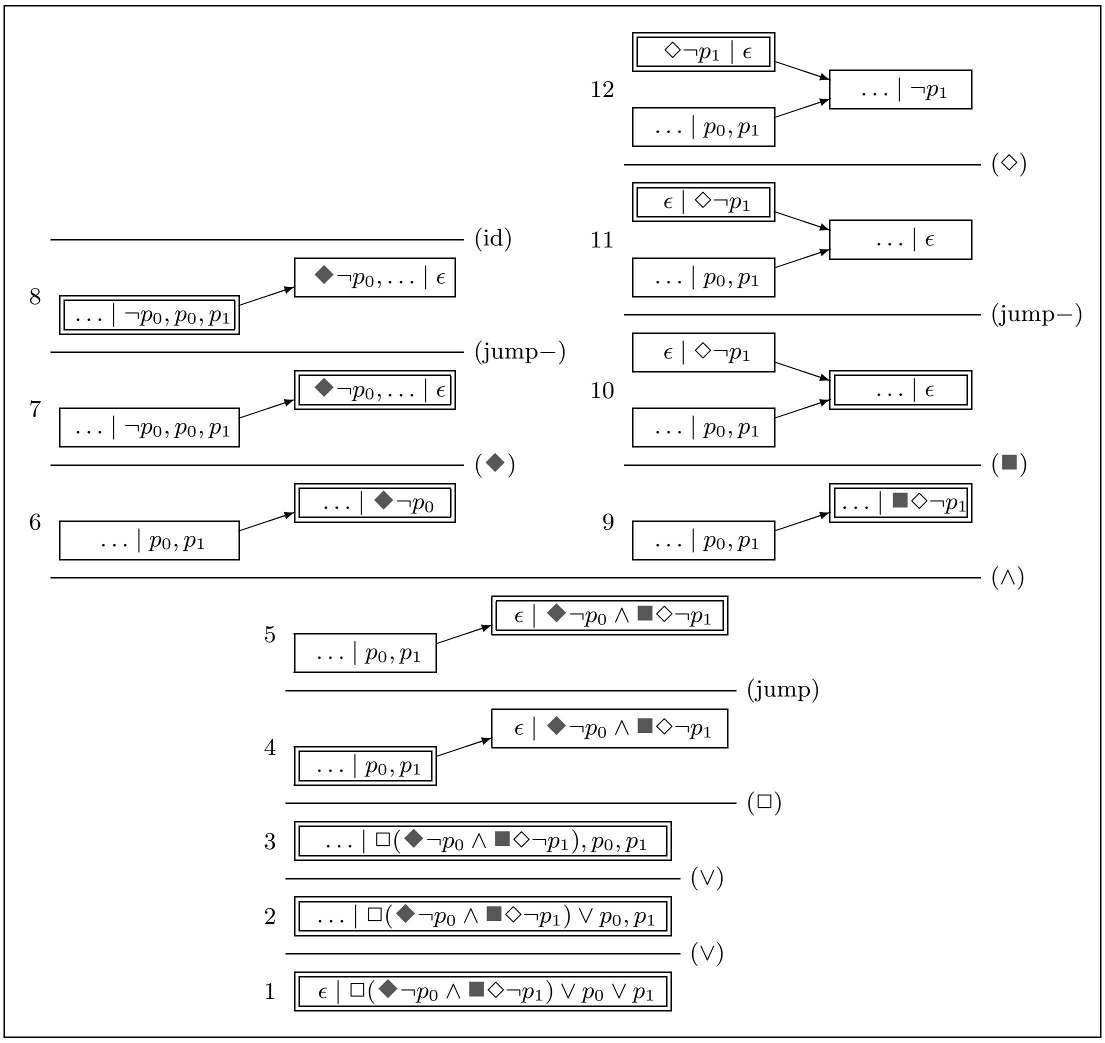

(id)
6 $\Box\blacklozenge\neg p_0, A \mid p_0, \neg p_0$
$\blacklozenge\neg p_0 \mid \epsilon$
(jump−)
5 $\Box\blacklozenge\neg p_0, A \mid p_0, \neg p_0$
$\blacklozenge\neg p_0 \mid \epsilon$
($\blacklozenge$)
4 $\Box\blacklozenge\neg p_0, A \mid p_0$
$\epsilon \mid \blacklozenge\neg p_0$
(jump)
3 $\Box\blacklozenge\neg p_0, A \mid p_0$
$\epsilon \mid \blacklozenge\neg p_0$
($\Box$)
2 $A \mid \Box\blacklozenge\neg p_0, p_0$
($\vee$)
1 $\epsilon \mid \Box\blacklozenge\neg p_0 \vee p_0$

Figure 4.v: The proof in $\mathsf{K}_t^{\mathcal{G}}$ from the first example in 4.7.5. See 4.u for the corresponding search tree.



The search starts at node 1. After two backward applications of ($\vee$) and creating a new world we jump into this new world (node 5). Then we apply ($\wedge$) backwards. One branch ends in an instance of an axiom (node 8), since a $\neg p_0$ is written back into the root. The other branch fails.
Therefore the K_t graph $\boxed{\epsilon \mid \Box(\blacklozenge\neg p_0 \wedge \blacksquare\Diamond\neg p_1) \vee p_0 \vee p_1}$ is not provable in $\mathsf{K}_t^{\mathcal{G}}$.

Figure 4.w: The search tree in $\mathsf{K}_t^{\mathcal{G}}$ of the second example in remark 4.7.5. See figure 4.x for the corresponding non-proof and 4.y for the resulting countermodel.



12
$\Diamond\neg p_1 \mid \epsilon$
$\ldots \mid p_0, p_1$
$\ldots \mid \neg p_1$
$(\Diamond)$
11
$\epsilon \mid \Diamond\neg p_1$
$\ldots \mid p_0, p_1$
$\ldots \mid \epsilon$
(jump−)
10
$\epsilon \mid \Diamond\neg p_1$
$\ldots \mid p_0, p_1$
$\ldots \mid \epsilon$
$(\blacksquare)$
9
$\ldots \mid p_0, p_1$
$\ldots \mid \blacksquare\Diamond\neg p_1$
(id)
8
$\ldots \mid \neg p_0, p_0, p_1$
$\blacklozenge\neg p_0, \ldots \mid \epsilon$
(jump−)
7
$\ldots \mid \neg p_0, p_0, p_1$
$\blacklozenge\neg p_0, \ldots \mid \epsilon$
$(\blacklozenge)$
6
$\ldots \mid p_0, p_1$
$\ldots \mid \blacklozenge\neg p_0$
$(\wedge)$
5
$\ldots \mid p_0, p_1$
$\epsilon \mid \blacklozenge\neg p_0 \wedge \blacksquare\Diamond\neg p_1$
(jump)
4
$\ldots \mid p_0, p_1$
$\epsilon \mid \blacklozenge\neg p_0 \wedge \blacksquare\Diamond\neg p_1$
$(\Box)$
3
$\ldots \mid \Box(\blacklozenge\neg p_0 \wedge \blacksquare\Diamond\neg p_1), p_0, p_1$
$(\vee)$
2
$\ldots \mid \Box(\blacklozenge\neg p_0 \wedge \blacksquare\Diamond\neg p_1) \vee p_0, p_1$
$(\vee)$
1
$\epsilon \mid \Box(\blacklozenge\neg p_0 \wedge \blacksquare\Diamond\neg p_1) \vee p_0 \vee p_1$

Figure 4.x: The non-proof in $\mathsf{K}_t^{\mathcal{G}}$ of the second example in remark 4.7.5. See figure 4.w for the corresponding search tree and figure 4.y for the resulting countermodel.

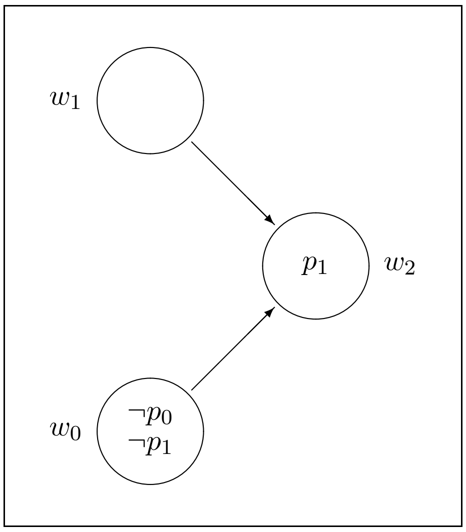


Figure 4.y: The countermodel of the example in remark 4.7.5. See figure 4.x for the corresponding non-proof.

4.8.2 REMARK PLTL

In [Gou89] and [Wol85] tableau calculi for PLTL are presented. These tableau calculi are similar to our graph calculi. Note that it consists of two phases: First a graph is generated, and then we have to check whether the so-called eventualities are satisfied. This is an important difference between PLTL and the modal logics we consider. The ($\Box$) rules in our graph calculi make sure that if $\Box A$ occurs in a world, then A will occur in a successor world.

4.8.3 REMARK CTL

See [Eme90] for a tableau calculus for CTL.

4.8.4 REMARK other logics

See section 5.8 for sequent and tableaux calculi for other logics. In many cases it is obvious how to obtain the corresponding graph calculus.

4.9 SUMMARY

We have defined graph calculi for the logics K, $\mathsf{K}+T$, KT, $\mathsf{KT}+T$, $\mathsf{S4}$, K_t. A proof search in these calculi can be seen as a search for a countermodel, and we have proved the equivalence with the possible world semantics in this way.

For all these logics there is a graph calculus where all rules are invertible, and thus no backtracking is required during backward proof search in these calculi. If the search fails, then a countermodel can be read off immediately from a failing branch.

With the exception of K_t, we have defined a second graph calculus for each logic which contains a non-invertible rule. We thus have to use backtracking during backward proof search. These calculi will be used in the next chapter to obtain sequent calculi in a perspicuous way.

5

SEQUENT CALCULI

Before him stood a wide dark arch opening into three passages: all led in the same general direction, eastwards; but the left-hand passage plunged down, while the right-hand climbed up, and the middle way seemed to run on, smooth and level but very narrow.
'I have no memory of this place at all!' said Gandalf, standing uncertainly under the arch.

J.R.R. Tolkien. The Lord of the Rings.

5.1 INTRODUCTION

5.1.1 REMARK from graph calculi to sequent calculi

When doing backward proof search in a graph calculus, we have a graph at each node of the search tree. An implementation of such a proof search has two disadvantages: The data structures are complicated and the space complexity is large (cp. chapter 6).

In this chapter we convert the graph calculi from the previous chapter into sequent calculi. A sequent consists of two multisets of formulas, i.e. it is a much simpler structure than a graph. Moreover, we will see in chapter 6 that the space complexity is also improved.

5.1.2 DEFINITION sequent, sequent scheme

We consider sequents with and without histories. An L sequent without a history is a pair of multisets of formulas of L, whereas an L sequent with a history has an additional history. A history is either a multiset of formulas or a multiset of multisets of formulas. Analogously, we define sequent schemes, using multiset schemes instead of multisets.

5.1.3 DEFINITION sequent calculus

A sequent calculus for the logic L consists of the following parts:

- A finite set of axioms.

  An axiom has the form $\frac{}{S}$, where S is a sequent scheme of L.
- A finite set of rules.

  A rule has the form $\frac{S_1}{S}$ or $\frac{S_1 \quad S_2}{S}$, where S_1, S_2, S are sequent schemes of L. Sometimes there is an additional condition.

All sequent schemes in a sequent calculus must have the same structure. Instances of axioms and rules are defined 'as usual'.

Example

See 5.2.2 for examples of axioms and rules. The proof in the example of 5.2.2 consists of instances of axioms and rules. Each sequent of this calculus consists of two multisets of formulas.

5.1.4 DEFINITION $L^{\mathcal{S}} \vdash S$

Let $L^{\mathcal{S}}$ be a sequent calculus for the logic L. We inductively define $L^{\mathcal{S}} \vdash S$ for sequents S:

- If $\frac{}{S_1}$ is an instance of an axiom of $L^{\mathcal{S}}$, then $L^{\mathcal{S}} \vdash S_1$.
- If $\frac{S_1}{S}$ instance of a rule of $L^{\mathcal{S}}$ and $L^{\mathcal{S}} \vdash S_1$, then $L^{\mathcal{S}} \vdash S$.
- If $\frac{S_1 \quad S_2}{S}$ instance of a rule of $L^{\mathcal{S}}$ and $L^{\mathcal{S}} \vdash S_1$, $L^{\mathcal{S}} \vdash S_2$, then $L^{\mathcal{S}} \vdash S$.

If $L^{\mathcal{S}} \vdash S$, then we say that S is provable in $L^{\mathcal{S}}$.

5.1.5 DEFINITION depth of a proof

Let $\mathcal{P}$ be a proof of the sequent S in the sequent calculus $L^{\mathcal{S}}$. If $\mathcal{P}$ is an instance of an axiom, then the depth of $\mathcal{P}$ is 0. If the last step in $\mathcal{P}$ is $\frac{S_1}{S}$, then the depth of $\mathcal{P}$ is the depth of the proof of S_1 plus one. If the last step in $\mathcal{P}$ is $\frac{S_1 \quad S_2}{S}$, and d_1 resp. d_2 are the depths of the proofs of S_1 and S_2, then the depth of $\mathcal{P}$ is $1 + \max(\{d_1, d_2\})$.

5.1.6 REMARK two-sided versus one-sided

We discuss only one-sided sequent calculi in this chapter. This facilitates the discussion and makes it easier to point out the important issues.

The transformation from one-sided to two-sided calculi is the same for all the logics we consider. Also, all optimisations work both in the one-sided and in the two-sided case.

One difference that concerns backward proof search is the (id) axiom. The (id) axiom of the calculus $\mathsf{K}^{\mathcal{S}}$, which is defined later on, is $\Diamond\Sigma \mid P, \neg P, \Gamma$. In the two-sided case we could use the axiom $\Box\Pi \mid A, \Delta \supset \Diamond\Sigma \mid A, \Gamma$, which is applicable more often. For example, take the sequent $\epsilon \mid \Box p_0 \supset \epsilon \mid \Box\Pi, \Box p_0$. This is an instance of the two-sided axiom, i.e. backward proof search immediately succeeds. The corresponding one-sided sequent $\epsilon \mid \Diamond\neg p_0, \Box\Pi, \Box p_0$ is not an axiom of $\mathsf{K}^{\mathcal{S}}$, and a lot of time can be lost during backward proof search until the ($\Box$) rule is applied backwards on $\Box p_0$. Such situations occur for example when checking the instances of figure 11.3.9. This is not a real problem as for example theorem 5.2.16 shows.

Another advantage of two-sided calculi is the possibility of special rules for equivalences. This means that during backward proof, search equivalences are split up on the fly and not at the beginning as in the case of our one-sided sequent calculi.

5.1.7 REMARK weakening rules

Assume that $\mathcal{P}$ is a proof of a sequent S in the usual sequent calculus for classical propositional logic. If we add a formula A to every sequent in the proof, then we obtain a proof of the sequent S plus A.

In the case of modal logic, the situation is a bit more complicated. In the one-sided sequent calculus $\mathsf{K}^{\mathcal{S}}$, which is defined in the following section, we can replace the axiom $\Diamond\Sigma \mid P, \neg P, \Gamma$ by the axiom $\Diamond\Sigma \mid P, \neg P$, the axiom $\Diamond\Sigma \mid \mathsf{true}, \Gamma$ by the axiom $\Diamond\Sigma \mid \mathsf{true}$, and the rule

$$\frac{\epsilon|A,\Sigma}{\Diamond\Sigma|\Box A,\Gamma}(\Box) \quad \text{by} \quad \frac{\epsilon|A,\Sigma}{\Diamond\Sigma|\Box A}(\Box')$$

if we add the rule

$$\frac{\Diamond\Sigma \mid \Gamma}{\Diamond\Sigma \mid A, \Gamma}\,(\text{weak})$$

to the calculus. Note that now $(\Box')$ is an invertible rule (the definition of 'invertible rule' follows below), but (weak) is not. The same method can be applied to the other sequent calculi. In this thesis we do not use this variant because we want to use the calculi for backward proof search. However, we will prove the admissibility of weakening for most of our sequent calculi.

5.1.8 DEFINITION (strongly) invertible rule

A rule with one premise of a sequent calculus $L^{\mathcal{S}}$ is invertible iff for all instances $\frac{S_1}{S}$ of this rule:

$$L^{\mathcal{S}} \vdash S_1 \;\Leftrightarrow\; L^{\mathcal{S}} \vdash S$$

The rule is strongly invertible if we have in addition: If there exists a proof of S with depth d, then there exists a proof of S_1 whose depth is at most d.

A rule with two premises of a sequent calculus $L^{\mathcal{S}}$ is invertible iff for all instances $\frac{S_1 \quad S_2}{S}$ of this rule:

$$L^{\mathcal{S}} \vdash S_1 \text{ and } L^{\mathcal{S}} \vdash S_2 \;\Leftrightarrow\; L^{\mathcal{S}} \vdash S$$

The rule is strongly invertible if we have in addition: If there exists a proof of S with depth d, then there exist proofs of S_1 and S_2 whose depth is at most d.

5.1.9 REMARK invertible rules

The invertibility of rules is very important for backward proof search in sequent calculi: As long as we apply invertible rules backwards, we do not have to backtrack. See also remark 4.2.7.

5.1.10 DEFINITION permutation of rules

Let $L^{\mathcal{S}}$ be a sequent calculus. We assume that x is an axiom of this sequent calculus, r_1, s_1 are two rules of $L^{\mathcal{S}}$ with one premise, and r_2, s_2 are two rules of $L^{\mathcal{S}}$ with two premises.

- Meaning of $x \rightsquigarrow s_1 x$:
  For all sequents S: For all variables P: For all formulas C: For all proofs $\mathcal{P}$ in $L^{\mathcal{S}}$: If $\mathcal{P}$ is of the form $\frac{}{S}x$ with main formulas P, $\neg P$ and the rule s_1 is applicable backwards on S with main formula C, then there exists a proof of the form

$$\cfrac{\cfrac{}{S'}\,x}{S}\,s_1$$

such that the main formulas in the instance of the axiom x are P, $\neg P$ and the main formula in the application of s_1 is C.

- Meaning of $x \rightsquigarrow s_2 x$:
  For all sequents S: For all variables P: For all formulas C: For all proofs $\mathcal{P}$ in $L^{\mathcal{S}}$: If $\mathcal{P}$ is of the form $\frac{}{S}x$ with main formulas P, $\neg P$ and the rule s_2 is applicable backwards on S with main formula C, then there exists a proof of the form

$$\cfrac{\cfrac{}{S_1'}\,x \qquad \cfrac{}{S_2'}\,x}{S}\,s_2$$

  such that the main formulas in the two instances of x are P, $\neg P$ and the main formula in the application of s_2 is C.

- Meaning of $r_1 \cdots \rightsquigarrow s_1 r_1 \cdots$: For all sequents S: For all formulas B, C: For all proofs $\mathcal{P}$ in $L^{\mathcal{S}}$: If $\mathcal{P}$ is of the form $\frac{\vdots}{S}r_1$ where the main formula in the r_1 application is B and the rule s_1 is applicable backwards on S with main formula C, then there exists a proof of the form

$$\cfrac{\cfrac{\vdots}{S'}\,r_1}{S}\,s_1$$

  such that the main formula in the application of r_1 is B, the main formula in the application of s_1 is C, and the depth of the new proof is the depth of the original proof plus one.

- Meaning of $r_1 \cdots \rightsquigarrow s_2 r_1 \cdots$: For all sequents S: For all formulas B, C: For all proofs $\mathcal{P}$ in $L^{\mathcal{S}}$: If $\mathcal{P}$ is of the form $\frac{\vdots}{S}r_1$ where the main formula in the r_1 application is B and the rule s_2 is applicable backwards on S with main formula C, then there exists a proof of the form

$$\cfrac{\cfrac{\vdots}{S_1'}\,r_1 \qquad \cfrac{\vdots}{S_2'}\,r_1}{S}\,s_2$$

  such that the main formula in the two applications of r_1 is B, the main formula in the application of s_2 is C, and the depth of the new proof is the depth of the original proof plus one.

- Meaning of $r_2 s_2 \cdots \rightsquigarrow s_2 r_2 \cdots$: For all sequents S: For all formulas B, C: For all proofs $\mathcal{P}$ in $L^{\mathcal{S}}$: If $\mathcal{P}$ is of the form

$$\cfrac{\cfrac{\vdots \qquad \vdots}{S_1'}\,s_2 \qquad \cfrac{\vdots \qquad \vdots}{S_2'}\,s_2}{S}\,r_2$$

  where the main formula in the r_2 application is B and the main formula in the two s_2 applications is C, then there exists a proof of the form

$$\cfrac{\cfrac{\vdots \qquad \vdots}{S_1''}\,r_2 \qquad \cfrac{\vdots \qquad \vdots}{S_2''}\,r_2}{S}\,s_2$$

  such that the main formula in the two applications of r_2 is B, the main formula in the application of s_2 is C, and the new proof is not deeper than the original one. Moreover, if $\frac{S_1' \quad S_2'}{S}$ is an instance of r_2 and s_2 is applicable backwards on S with main formula C, then s_2 is applicable backwards both on S_1' and S_2' with main formula C (provided that the instance of r_2 is not an application of s_2 with main formula C).

- Meaning of $r_1 s_2 \cdots \rightsquigarrow s_2 r_1 \cdots$: Analogous to $r_2 s_2 \cdots \rightsquigarrow s_2 r_2 \cdots$, with the following transformation:

$$\dfrac{\dfrac{\vdots}{S'}\,s_2}{S}\,r_1 \qquad \rightsquigarrow \qquad \dfrac{\dfrac{\vdots}{S_1''}\,r_1 \qquad \dfrac{\vdots}{S_2''}\,r_1}{S}\,s_2$$

- Meaning of $r_2 s_1 \cdots \rightsquigarrow s_1 r_2 \cdots$: Analogous to $r_2 s_2 \cdots \rightsquigarrow s_2 r_2 \cdots$, with the following transformation:

$$\dfrac{\dfrac{\vdots}{S_1'}\,s_1 \qquad \dfrac{\vdots}{S_2'}\,s_1}{S}\,r_2 \qquad \rightsquigarrow \qquad \dfrac{\dfrac{\vdots \qquad \vdots}{S''}\,r_2}{S}\,s_1$$

- Meaning of $r_1 s_1 \cdots \rightsquigarrow s_1 r_1 \cdots$: Analogous to $r_2 s_2 \cdots \rightsquigarrow s_2 r_2 \cdots$, with the following transformation:

$$\dfrac{\dfrac{\vdots}{S'}\,s_1}{S}\,r_1 \qquad \rightsquigarrow \qquad \dfrac{\dfrac{\vdots}{S''}\,r_1}{S}\,s_1$$

5.1.11 THEOREM strongly invertible rule

In order to prove the strong invertibility of a rule r in a sequent calculus $L^{\mathcal{S}}$ it is sufficient to show:

- for all axioms x of $L^{\mathcal{S}}$: $x \rightsquigarrow rx$
- for all rules s of $L^{\mathcal{S}}$: $sr \cdots \rightsquigarrow rs \cdots$ or $s \cdots \rightsquigarrow rs \cdots$

Proof

Let r be a rule of $L^{\mathcal{S}}$ that satisfies the two conditions above and $\mathcal{P}$ a proof of a sequent S. We assume that the rule r is applicable backwards on S with main formula C. With an induction on proof depth, we prove that there exists a proof $\mathcal{P}'$ of S in $L^{\mathcal{S}}$ that ends with an application of r with main formula C and whose depth is at most the depth of $\mathcal{P}$ plus one.

First we assume that r is a rule with one premise.

1. $\mathcal{P}$ is of the form $\frac{}{S}x$, i.e. x is an axiom:
   Since we have $x \rightsquigarrow rx$ we know that there exists a proof of the form

$$\dfrac{\dfrac{}{S'}\,x}{S}\,r$$

   where the main formula of the r application is C. This proof has depth one, i.e. the depth of $\mathcal{P}$ plus one.

2. $\mathcal{P}$ is of the form $\frac{\vdots}{S}s$, d the depth of $\mathcal{P}$, and $s \cdots \rightsquigarrow rs \cdots$:
   From $s \cdots \rightsquigarrow rs \cdots$ follows immediately the existence of a proof with depth $\leq d + 1$ and an application of r with main formula C as its last step.

3. $\mathcal{P}$ is of the form $\dfrac{\overset{\vdots}{S'}}{S}s$, d the depth of $\mathcal{P}$, and $sr\cdots \rightsquigarrow rs\cdots$:
   Since $sr\cdots \rightsquigarrow rs\cdots$ we know that r is applicable backwards on S' with main formula C. Thus there exists a proof of S' with depth $\leq d$ and an application of r with main formula C as its last step because of the induction hypothesis. We append the application of s and obtain a proof with depth $\leq d+1$. With $sr\cdots \rightsquigarrow rs\cdots$ follows the existence of a proof of S with depth $\leq d+1$ and an application of r with main formula C as its last step.

4. $\mathcal{P}$ is of the form $\dfrac{\overset{\vdots}{S'_1}\quad\overset{\vdots}{S'_2}}{S}s$, d the depth of $\mathcal{P}$, and $sr\cdots \rightsquigarrow rs\cdots$:
   Since $sr\cdots \rightsquigarrow rs\cdots$ we know that r is applicable backwards both on S'_1 and S'_2 with main formula C. Thus there exist proofs of S'_1 and S'_2 with depth $\leq d$ and an application of r with main formula C as their last step because of the induction hypothesis. We append the application of s and obtain a proof of S with depth $\leq d+1$. With $sr\cdots \rightsquigarrow rs\cdots$ follows the existence of a proof with depth $\leq d+1$ and an application of r with main formula C as its last step.

The strong invertibility of r follows immediately. The cases where the rule r has two premises are analogous.

See remark 5.2.5 for an example.

5.1.12 REMARK construct Hilbert-style proofs

We show how to construct Hilbert-style calculus proofs from proofs in the usual sequent calculus. Note that the resulting proofs are by no means short or elegant.

5.1.13 THEOREM subformula property

Let A be a formula in negation normal form. If $\mathcal{P}$ is a proof of a formula A in one of the following one-sided sequent calculi without theories defined in this chapter, A is in negation normal form, and B is a formula in a sequent of this proof $\mathcal{P}$, then $B \in \text{subfmls}(A)$.

If a theory T is involved, then $B \in \text{subfmls}(A)$ or $B \in \text{subfmls}(\text{nnf}(\neg C))$ for some $C \in T$.

5.1.14 REMARK termination

For some of our sequent calculi, backward proof search terminates automatically. For the other calculi, we will show that they satisfy the subformula property and we can always eliminate duplicate formulas. Then only a finite number of 'different' sequents can occur during backward proof search. Consequently we can always ensure the termination of backward proof search with a loop-check. If a sequent occurred already before on the same branch, then we stop. We will show that in the case of our sequent calculi, it is correct to return 'fail' when a loop is detected. This is different for example for PLTL, where a loop does not always mean that a countermodel has been found.

5.1.15 REMARK termination and search strategy

Assume that only a finite number of 'different' sequents can occur during backward proof search. If we use a depth-first search, then it is possible that the search does not terminate even in the case of provable formulas. With a breadth-first search, we always find a proof if there is one. However, if the formula is not provable, then — independent of the search strategy — we cannot be sure that the search terminates.

5.1.16 REMARK adding a loop-check to a sequent calculus

Assume that backward proof search in a sequent calculus $L^{\mathcal{S}}$ does not terminate, that the calculus satisfies the subformula property, and that we can eliminate duplicate formulas.

We add a so-called history H as a third multiset to the sequents. This history contains the required information about the previous worlds.

In the simplest solution, the history contains all the sequents that occurred on this branch in the past. If we take the calculus $(\mathsf{K}+T)^{\mathcal{S}}$ defined later on, then the new $(\vee)$ rule for example becomes $\frac{H,[\Diamond\Sigma,A\vee B,\Gamma]|\Diamond\Sigma|A,B,\Gamma}{H|\Diamond\Sigma|A\vee B,\Gamma}$ with the condition $[\Diamond\Sigma, A\vee B,\Gamma]\notin H$. Thus $\frac{[p_0\vee\Box p_1,p_3],[p_0\vee\Box p_1]|\epsilon|p_0,\Box p_1,p_3}{[p_0\vee\Box p_1]|\epsilon|p_0\vee\Box p_1,p_3}$ is an instance of the $(\vee)$ rule, but $\frac{[p_0\vee\Box p_1,p_3],[p_0\vee\Box p_1,p_3]|\epsilon|p_0,\Box p_1,p_3}{[p_0\vee\Box p_1,p_3]|\epsilon|p_0\vee\Box p_1,p_3}$ is not. Clearly, backward proof search in this new calculus terminates provided that we always eliminate duplicate formulas immediately. However, with this method one obtains rather large histories and thus an inefficient loop-check. Often it is possible to put only some sequents into the history without losing the termination property.

5.1.17 REMARK extend the identity axiom

The identity axioms in our sequent calculi without a history are of the form $\Diamond\Sigma \mid P,\neg P,\Gamma$. We will show that the restriction to variables is not necessary, i.e. that all sequents of the form $\Diamond\Sigma \mid A,\neg A,\Gamma$ are provable (cp. remark 5.1.6).

5.1.18 REMARK contraction, duplicate formulas

In the sequent calculus $\mathsf{K}^{\mathcal{S},2}$ the contraction rule $\frac{\Diamond\Sigma|A,A,\Gamma}{\Diamond\Sigma|A,\Gamma}$ is admissible, but not necessary. This rule allows us to remove duplicate formulas from sequents during backward proof search in $\mathsf{K}^{\mathcal{S},2}$. Analogous results can be stated for all our sequent calculi. The situation is different for example in linear logic, where for example $(p_0\otimes p_0),\neg p_0,\neg p_0$ is provable, but $(p_0\otimes p_0),\neg p_0$ is not.

5.1.19 THEOREM use-check in CPC

Let $\mathsf{CPC}^{\mathcal{S}}$ be the usual one-sided sequent calculus for classical propositional logic, i.e. the calculus with the two axioms $\overline{P,\neg P,\Gamma}$ (id) and $\overline{\mathsf{true},\Gamma}$ (true), and the two rules $\frac{A,B,\Gamma}{A\vee B,\Gamma}$ $(\vee)$ and $\frac{A,\Gamma \quad B,\Gamma}{A\wedge B,\Gamma}$ $(\wedge)$. The main formulas are P and $\neg P$ in (id), true in (true), $A\vee B$ in $(\vee)$, and $A\wedge B$ in $(\wedge)$. The side formulas both of $(\vee)$ and $(\wedge)$ are A and B.

Let $\mathcal{P}'$ be a proof of a sequent S in $\mathsf{CPC}^{\mathcal{S}}$. $\mathcal{P}'$ is a marked proof if:

- The main formulas of all instances of axioms are marked.
- If in an instance of a rule, a side formula is marked, then the main formula is also marked.
- If in the premise of an instance of a rule, a formula in Γ is marked, then the corresponding formula in the multiset Γ in the conclusion is marked.

Let $\mathcal{P}$ be a marked proof of the sequent A,Γ in the calculus $\mathsf{CPC}^{\mathcal{S}}$. If this occurrence of A is not marked, then $\mathsf{CPC}\vdash\Gamma$. With weakening, we obtain $\mathsf{CPC}\vdash A\wedge B,\Gamma$.

Example

In figure 5.a, the subproofs of the sequents $p_3,p_0,p_0,\neg p_0\wedge p_1,\neg p_0\wedge\neg p_1$ and $p_3,p_1,p_0\wedge\neg p_1,\neg p_0\wedge p_1,\neg p_0\wedge\neg p_1$ are marked proofs.

5.1.20 REMARK use-check to prune the search tree

During backward proof search, use-check can help to prune the search tree. At each branching in the search tree caused by the $(\wedge)$ rule, we check after having proved one branch, whether we can cut off the second branch.

Example

We check with backward proof search in the usual one-sided sequent calculus for CPC whether $A \equiv (p_3 \wedge p_4) \vee (p_0 \wedge p_1) \vee (p_0 \wedge \neg p_1) \vee (\neg p_0 \wedge p_1) \vee (\neg p_0 \wedge \neg p_1)$ is valid. We do a depth-first search, doing the left premise first. We use a bold font for the 'used' formulas, and branches we did not consider due to use-check are marked with *cut off* (see figure 5.a).

Since the second p_0 is not used in the first branch of the proof of $p_3, p_0, p_0 \wedge \neg p_1, \neg p_0 \wedge p_1, \neg p_0 \wedge \neg p_1$, we can cut off the second branch above this sequent. Here use-check is applicable because we applied $(\wedge)$ on the 'wrong' conjunction of the sequent. Use-check can help to recognise such wrong choices after having done the first branch. Note that $p_0 \wedge \neg p_1$ is not superfluous in A, but just in this part of the search tree.

We cannot cut off the second branch in the proof of $p_3, p_0 \wedge p_1, p_0 \wedge \neg p_1, \neg p_0 \wedge p_1, \neg p_0 \wedge \neg p_1$, since the first p_0 was used in the first branch.

However, we can again cut off a branch right at the beginning, since that occurrence of p_3 was not used on the left hand side. Here, the conjunction $p_3 \wedge p_4$ is really superfluous.

The power of use-check becomes fully visible in the case of longer formulas. An extreme example is (as usual) the pigeonhole formulas.

5.1.21 REMARK more about use-check

The term 'use-check' was used in [SFH92] for backward proof search in sequent calculi for IPC. Proof condensation, defined in [OS88] for CPC, is just another term for the same method.

There are several other methods that have similar effects as use-check:

- For CPC and propositional many-valued logics, the so-called lemma generation was proposed (see [d'A92], [Häh93]). There the usual $(\wedge)$ rule is replaced by $\frac{A,\Gamma \quad \neg A, B, \Gamma}{A \wedge B, \Gamma}(\wedge')$. Lemma generation and use-check seem to have similar effects in the case of CPC; however, lemma generation is much easier to implement and describe. In the case of modal logics there seems to be a considerable difference (see the remarks 6.3.3 and 6.5.6).
- In [Gou89] a variant of lemma generation was used for tableaux for PLTL.
- Also the connection method (see for example [BE93], [Wal90]) can be seen as a method to avoid 'useless' branchings. The advantage of the connection method is that we do not have to start at the top level connectives as when doing backward proof search in a sequent calculus, but wherever we want. In this way unnecessary branchings can be avoided.

5.2 K

5.2.1 REMARK from $\mathsf{K}^{\mathcal{G},2}$ to $\mathsf{K}^{\mathcal{S}}$

In the graph calculus $\mathsf{K}^{\mathcal{G},2}$ no rule influences a node that is nearer to the root than the actual node; especially there is no (jump−) rule. Thus we can forget everything about the worlds 'below' the actual world.

5.2.2 DEFINITION sequent calculus $\mathsf{K}^{\mathcal{S}}$

axioms:

$$\frac{}{\Diamond\Sigma \mid \mathsf{true}, \Gamma}\ (\mathsf{true}) \qquad\qquad \frac{}{\Diamond\Sigma \mid P, \neg P, \Gamma}\ (\mathrm{id})$$

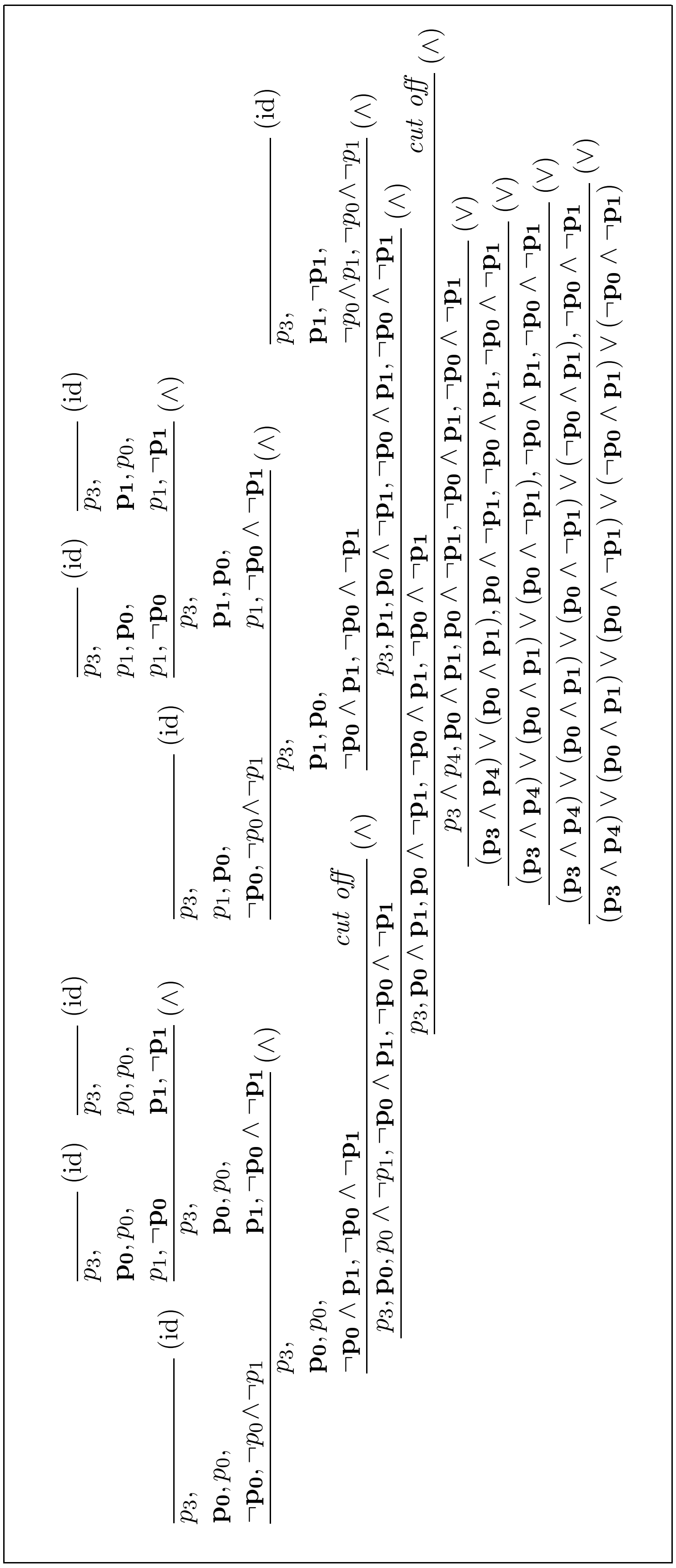


Figure 5.a: The backward proof search in CPC of the example in 5.1.20. Use-check helps to cut off branches.

rules:

$$\frac{\Diamond\Sigma \mid A, B, \Gamma}{\Diamond\Sigma \mid A \vee B, \Gamma}\ (\vee) \qquad \frac{\Diamond\Sigma \mid A, \Gamma \quad \Diamond\Sigma \mid B, \Gamma}{\Diamond\Sigma \mid A \wedge B, \Gamma}\ (\wedge)$$

$$\frac{\Diamond A, \Diamond\Sigma \mid \Gamma}{\Diamond\Sigma \mid \Diamond A, \Gamma}\ (\Diamond) \qquad \frac{\epsilon \mid A, \Sigma}{\Diamond\Sigma \mid \Box A, \Gamma}\ (\Box)$$

main formulas: true in (true), P and $\neg P$ in (id), $A \vee B$ in ($\vee$), $A \wedge B$ in ($\wedge$), $\Diamond A$ in ($\Diamond$), $\Box A$ in ($\Box$)

Example

$\mathsf{K}^{\mathcal{S}} \vdash \epsilon \mid \Diamond(p_0 \wedge \neg p_1) \vee (\Diamond\neg p_0 \vee \Box p_1)$, as the following proof shows.

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{}{\epsilon \mid p_1, p_0, \neg p_0}\ (\text{id}) \quad \cfrac{}{\epsilon \mid p_1, \neg p_1, \neg p_0}\ (\text{id})}{\epsilon \mid p_1, p_0 \wedge \neg p_1, \neg p_0}\ (\wedge)}{\Diamond(p_0 \wedge \neg p_1), \Diamond\neg p_0 \mid \Box p_1}\ (\Box)}{\Diamond(p_0 \wedge \neg p_1) \mid \Diamond\neg p_0, \Box p_1}\ (\Diamond)}{\epsilon \mid \Diamond(p_0 \wedge \neg p_1), \Diamond\neg p_0, \Box p_1}\ (\Diamond)}{\epsilon \mid \Diamond(p_0 \wedge \neg p_1), \Diamond\neg p_0 \vee \Box p_1}\ (\vee)}{\epsilon \mid \Diamond(p_0 \wedge \neg p_1) \vee (\Diamond\neg p_0 \vee \Box p_1)}\ (\vee)}{}$$

5.2.3 THEOREM $\mathsf{K}^{\mathcal{S}}$: weakening

If $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid \Gamma$, then $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$. Moreover, if d is the depth of the proof of $\Diamond\Sigma \mid \Gamma$, then there exists a proof of $\Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$ whose depth is at most d.

Proof

We make an induction on proof depth. The only interesting case is the ($\Box$) rule.

1. The last step is $\frac{\Diamond\Sigma|A,B,\Gamma}{\Diamond\Sigma|A\vee B,\Gamma}(\vee)$: We obtain $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, A, B, \Gamma$ with the induction hypothesis. With a ($\vee$) application follows $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, A \vee B, \Gamma$.
2. The last step is $\frac{\Diamond\Sigma|A,\Gamma \quad \Diamond\Sigma|A,\Gamma}{\Diamond\Sigma|A\wedge B,\Gamma}(\wedge)$: We obtain $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, A, \Gamma$ and $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, B, \Gamma$ with the induction hypothesis. With a ($\wedge$) application follows $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, A \wedge B, \Gamma$.
3. The last step is $\frac{\Diamond A,\Diamond\Sigma|\Gamma}{\Diamond\Sigma|\Diamond A,\Gamma}(\Diamond)$: We obtain $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Delta, \Diamond A, \Diamond\Sigma \mid \Pi, \Gamma$ with the induction hypothesis. With a ($\Diamond$) application follows $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, \Diamond A, \Gamma$.
4. The last step is $\frac{\epsilon|A,\Sigma}{\Diamond\Sigma|\Box A,\Gamma}(\Box)$: We obtain $\mathsf{K}^{\mathcal{S}} \vdash \epsilon \mid \Delta, A, \Sigma$ with the induction hypothesis. With a ($\Box$) application follows $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, \Box A, \Gamma$.

5.2.4 THEOREM $\mathsf{K}^{\mathcal{S}}$: invertible rules

The ($\Box$) rule of $\mathsf{K}^{\mathcal{S}}$ is not invertible. All the other rules are strongly invertible.

Proof

We can easily check the following statements for $\mathsf{K}^{\mathcal{S}}$:

1. (true) $\rightsquigarrow$ ($\vee$)(true), (id) $\rightsquigarrow$ ($\vee$)(id)

2. $(\mathsf{true}) \rightsquigarrow (\wedge)(\mathsf{true})$, $(\mathrm{id}) \rightsquigarrow (\wedge)(\mathrm{id})$
3. $(\mathsf{true}) \rightsquigarrow (\Diamond)(\mathsf{true})$, $(\mathrm{id}) \rightsquigarrow (\Diamond)(\mathrm{id})$
4. $(\vee)_1(\vee)_2 \cdots \rightsquigarrow (\vee)_2(\vee)_1 \cdots$, $(\wedge)(\vee) \cdots \rightsquigarrow (\vee)(\wedge) \cdots$, $(\Diamond)(\vee) \cdots \rightsquigarrow (\vee)(\Diamond) \cdots$, $(\Box) \cdots \rightsquigarrow (\vee)(\Box) \cdots$
5. $(\vee)(\wedge) \cdots \rightsquigarrow (\wedge)(\vee) \cdots$, $(\wedge)_1(\wedge)_2 \cdots \rightsquigarrow (\wedge)_2(\wedge)_1 \cdots$, $(\Diamond)(\wedge) \cdots \rightsquigarrow (\wedge)(\Diamond) \cdots$, $(\Box) \cdots \rightsquigarrow (\wedge)(\Box) \cdots$
6. $(\vee)(\Diamond) \cdots \rightsquigarrow (\Diamond)(\vee) \cdots$, $(\wedge)(\Diamond) \cdots \rightsquigarrow (\Diamond)(\wedge) \cdots$, $(\Diamond)_1(\Diamond)_2 \cdots \rightsquigarrow (\Diamond)_2(\Diamond)_1 \cdots$, $(\Box) \cdots \rightsquigarrow (\Diamond)(\Box) \cdots$

We only prove some typical cases. The sequent $\Diamond\Sigma \mid P, \neg P, A \vee B, \Gamma$ is an instance of the axiom (id) (with main formulas P and $\neg P$). Also the rule $(\vee)$ is applicable backwards on $\Diamond\Sigma \mid P, \neg P, A \vee B, \Gamma$ (with main formula $A \vee B$). Because of

$$\dfrac{}{\Diamond\Sigma \mid P, \neg P, A \vee B, \Gamma}\,(\mathrm{id}) \quad \rightsquigarrow \quad \dfrac{\dfrac{}{\Diamond\Sigma \mid P, \neg P, A, B, \Gamma}\,(\mathrm{id})}{\Diamond\Sigma \mid P, \neg P, A \vee B, \Gamma}\,(\vee)$$

we have $(\mathrm{id}) \rightsquigarrow (\vee)(\mathrm{id})$. The transformation

$$\dfrac{\dfrac{\dfrac{\vdots}{\Diamond\Sigma \mid A, C, D, \Gamma}}{\Diamond\Sigma \mid A, C \vee D, \Gamma}\,(\vee) \quad \dfrac{\dfrac{\vdots}{\Diamond\Sigma \mid B, C, D, \Gamma}}{\Diamond\Sigma \mid B, C \vee D, \Gamma}\,(\vee)}{\Diamond\Sigma \mid A \wedge B, C \vee D, \Gamma}\,(\wedge) \quad \rightsquigarrow \quad \dfrac{\dfrac{\dfrac{\vdots}{\Diamond\Sigma \mid A, C, D, \Gamma} \quad \dfrac{\vdots}{\Diamond\Sigma \mid B, C, D, \Gamma}}{\Diamond\Sigma \mid A \wedge B, C, D, \Gamma}\,(\wedge)}{\Diamond\Sigma \mid A \wedge B, C \vee D, \Gamma}\,(\vee)$$

shows that $(\wedge)(\vee) \cdots \rightsquigarrow (\vee)(\wedge) \cdots$. Note that we assume that $(\vee)$ occurs on both branches with the same main formula. With the transformation

$$\dfrac{\dfrac{\dfrac{\vdots}{\Diamond\Sigma \mid A, B, C, D, \Gamma}}{\Diamond\Sigma \mid A, B, C \vee D, \Gamma}\,(\vee)}{\Diamond\Sigma \mid A \vee B, C \vee D, \Gamma}\,(\vee) \quad \rightsquigarrow \quad \dfrac{\dfrac{\dfrac{\vdots}{\Diamond\Sigma \mid A, B, C, D, \Gamma}}{\Diamond\Sigma \mid A \vee B, C, D, \Gamma}\,(\vee)}{\Diamond\Sigma \mid A \vee B, C \vee D, \Gamma}\,(\vee)$$

we obtain $(\vee)_1(\vee)_2 \cdots \rightsquigarrow (\vee)_2(\vee)_1 \cdots$. With the transformation

$$\dfrac{\dfrac{\vdots}{\epsilon \mid A, \Sigma}}{\Diamond\Sigma \mid \Box A, B \vee C, \Gamma}\,(\Box) \quad \rightsquigarrow \quad \dfrac{\dfrac{\dfrac{\vdots}{\epsilon \mid A, \Sigma}}{\Diamond\Sigma \mid \Box A, B, C, \Gamma}\,(\Box)}{\Diamond\Sigma \mid \Box A, B \vee C, \Gamma}\,(\vee)$$

we obtain $(\Box) \cdots \rightsquigarrow (\vee)(\Box) \cdots$, and finally we obtain $(\Box) \cdots \rightsquigarrow (\Diamond)(\Box) \cdots$ with the transformation below and weakening:

$$\dfrac{\epsilon \mid A, \Sigma}{\Diamond\Sigma \mid \Box A, \Diamond B, \Gamma}\,(\Box) \quad \rightsquigarrow \quad \dfrac{\dfrac{\epsilon \mid A, B, \Sigma}{\Diamond B, \Diamond\Sigma \mid \Box A, \Gamma}\,(\Box)}{\Diamond\Sigma \mid \Box A, \Diamond B, \Gamma}\,(\Diamond)$$

From 1. and 4. follows the strong invertibility of $(\vee)$, from 2. and 5. follows the strong invertibility of $(\wedge)$, and from 3. and 6. follows the strong invertibility of $(\Diamond)$.

Since $\mathsf{K}^{\mathcal{S}} \vdash \epsilon \mid \Box(p_0 \vee \neg p_0), \Box p_1$, but $\mathsf{K}^{\mathcal{S}} \nvdash \epsilon \mid p_1$, the $(\Box)$ rule is not invertible. Note that for example we do not have (id) $\leadsto$ $(\Box)$(id), since $\frac{}{\epsilon \mid p_0, \neg p_0, \Box p_1}$ is a proof of the sequent $\epsilon \mid p_0, \neg p_0, \Box p_1$ and there is no proof of this sequent with a $(\Box)$ application as its last step.

5.2.5 REMARK invertible rules

The proof of theorem 5.2.4 shows not only the existence of a proof, but together with the proof of theorem 5.1.11 we can in fact construct these proofs.

Example

Let $\mathcal{P}$ be the following proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\epsilon \mid p_0, \neg p_0}\,(\text{id})}{\Diamond\neg p_0 \mid \Box p_0, p_2, p_1, p_3}\,(\Box)}{\epsilon \mid \Box p_0, \Diamond\neg p_0, p_2, p_1, p_3}\,(\Diamond)}{\epsilon \mid \Box p_0, \Diamond\neg p_0 \vee p_2, p_1, p_3}\,(\vee)}{\epsilon \mid \Box p_0, \Diamond\neg p_0 \vee p_2, p_1 \vee p_3}\,(\vee) \qquad \dfrac{\dfrac{\dfrac{}{\epsilon \mid p_1, \neg p_1}\,(\text{id})}{\epsilon \mid p_1 \vee \neg p_1}\,(\vee)}{\epsilon \mid \Box(p_1 \vee \neg p_1), \Diamond\neg p_0 \vee p_2, p_1 \vee p_3}\,(\Box)}{\epsilon \mid \Box p_0 \wedge \Box(p_1 \vee \neg p_1), \Diamond\neg p_0 \vee p_2, p_1 \vee p_3}\,(\wedge)$$

Since $(\vee)$ is invertible we know that there exists a proof of $\epsilon \mid \Box p_0 \wedge \Box(p_1 \vee \neg p_1), \Diamond\neg p_0, p_2, p_1 \vee p_3$ in $\mathsf{K}^{\mathcal{S}}$. We obtain it as follows:

$$\mathcal{P} \quad \leadsto \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\epsilon \mid p_0, \neg p_0}\,(\text{id})}{\Diamond\neg p_0 \mid \Box p_0, p_2, p_1, p_3}\,(\Box)}{\epsilon \mid \Box p_0, \Diamond\neg p_0, p_2, p_1, p_3}\,(\Diamond)}{\epsilon \mid \Box p_0, \Diamond\neg p_0, p_2, p_1 \vee p_3}\,(\vee)}{\epsilon \mid \Box p_0, \Diamond\neg p_0 \vee p_2, p_1 \vee p_3}\,(\vee) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\epsilon \mid p_1, \neg p_1}\,(\text{id})}{\epsilon \mid p_1 \vee \neg p_1}\,(\vee)}{\epsilon \mid \Box(p_1 \vee \neg p_1), \Diamond\neg p_0, p_2, p_1 \vee p_3}\,(\Box)}{\epsilon \mid \Box(p_1 \vee \neg p_1), \Diamond\neg p_0 \vee p_2, p_1 \vee p_3}\,(\vee)}{\epsilon \mid \Box p_0 \wedge \Box(p_1 \vee \neg p_1), \Diamond\neg p_0 \vee p_2, p_1 \vee p_3}\,(\wedge)$$

$$\leadsto \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\epsilon \mid p_0, \neg p_0}\,(\text{id})}{\Diamond\neg p_0 \mid \Box p_0, p_2, p_1, p_3}\,(\Box)}{\epsilon \mid \Box p_0, \Diamond\neg p_0, p_2, p_1, p_3}\,(\Diamond)}{\epsilon \mid \Box p_0, \Diamond\neg p_0, p_2, p_1 \vee p_3}\,(\vee) \qquad \dfrac{\dfrac{\dfrac{}{\epsilon \mid p_1, \neg p_1}\,(\text{id})}{\epsilon \mid p_1 \vee \neg p_1}\,(\vee)}{\epsilon \mid \Box(p_1 \vee \neg p_1), \Diamond\neg p_0, p_2, p_1 \vee p_3}\,(\Box)}{\epsilon \mid \Box p_0 \wedge \Box(p_1 \vee \neg p_1), \Diamond\neg p_0, p_2, p_1 \vee p_3}\,(\wedge)}{\epsilon \mid \Box p_0 \wedge \Box(p_1 \vee \neg p_1), \Diamond\neg p_0 \vee p_2, p_1 \vee p_3}\,(\vee)$$

5.2.6 THEOREM $\mathsf{K}^{\mathcal{S}}$: duplicate formulas

$$\begin{aligned} \mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid A, A, \Gamma \quad &\Rightarrow \quad \mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid A, \Gamma \\ \mathsf{K}^{\mathcal{S}} \vdash \Diamond A, \Diamond A, \Diamond\Sigma \mid \Gamma \quad &\Rightarrow \quad \mathsf{K}^{\mathcal{S}} \vdash \Diamond A, \Diamond\Sigma \mid \Gamma \end{aligned}$$

Proof

We prove the two statements simultaneously with an induction on proof depth. We also prove that the depth of the new proof is at most the depth of the original one. First we consider the axioms:

1. $\Diamond\Sigma \mid A, A, \Gamma$ is an instance of the (true) axiom: Then $\Diamond\Sigma \mid A, \Gamma$ is also an instance of this axiom.
2. $\Diamond A, \Diamond A, \Diamond\Sigma \mid \Gamma$ is an instance of the (true) axiom: Then $\Diamond A, \Diamond\Sigma \mid \Gamma$ is also an instance of this axiom.
3. $\Diamond\Sigma \mid A, A, \Gamma$ is an instance of the (id) axiom: Then $\Diamond\Sigma \mid A, \Gamma$ is also an instance of this axiom.
4. $\Diamond A, \Diamond A, \Diamond\Sigma \mid \Gamma$ is an instance of the (id) axiom: Then $\Diamond A, \Diamond\Sigma \mid \Gamma$ is also an instance of this axiom.

Now we consider those cases where the duplicate formula is not the main formula in the last step in the proof:

1. The last step is $\frac{\Diamond\Sigma|A,A,B,C,\Gamma}{\Diamond\Sigma|A,A,B\vee C,\Gamma}(\vee)$: With the induction hypothesis and an application of $(\vee)$ we obtain $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid A, B \vee C, \Gamma$.
2. The last step is $\frac{\Diamond A,\Diamond A,\Diamond\Sigma|B,C,\Gamma}{\Diamond A,\Diamond A,\Diamond\Sigma|B\vee C,\Gamma}(\vee)$: Analogous to case 1.
3. The last step is $\frac{\Diamond\Sigma|A,A,B,\Gamma \quad \Diamond\Sigma|A,A,C,\Gamma}{\Diamond\Sigma|A,A,B\wedge C,\Gamma}(\wedge)$: With the induction hypothesis and an application of $(\wedge)$ we obtain $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid A, B \wedge C, \Gamma$.
4. The last step is $\frac{\Diamond A,\Diamond A,\Diamond\Sigma|B,\Gamma \quad \Diamond A,\Diamond A,\Diamond\Sigma|C,\Gamma}{\Diamond A,\Diamond A,\Diamond\Sigma|B\wedge C,\Gamma}(\wedge)$: Analogous to case 3.
5. The last step is $\frac{\Diamond B,\Diamond\Sigma|A,A,\Gamma}{\Diamond\Sigma|A,A,\Diamond B,\Gamma}(\Diamond)$: With the induction hypothesis and an application of $(\Diamond)$ we obtain $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid A, \Diamond B, \Gamma$.
6. The last step is $\frac{\Diamond B,\Diamond A,\Diamond A,\Diamond\Sigma|\Gamma}{\Diamond A,\Diamond A,\Diamond\Sigma|\Diamond B,\Gamma}(\Diamond)$: Analogous to case 5.
7. The last step in $\mathcal{P}$ is $\frac{\epsilon|B,\Sigma}{\Diamond\Sigma|A,A,\Box B,\Gamma}(\Box)$: With an application of $(\Box)$ we obtain $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid A, \Box B, \Gamma$.
8. The last step in $\mathcal{P}$ is $\frac{\epsilon|B,A,A,\Sigma}{\Diamond A,\Diamond A,\Diamond\Sigma|\Box B,\Gamma}(\Box)$: With the induction hypothesis we obtain $\mathsf{K}^{\mathcal{S}} \vdash \epsilon \mid B, A, \Sigma$, and with an application of $(\Box)$ follows $\mathsf{K}^{\mathcal{S}} \vdash \Diamond A, \Diamond\Sigma \mid \Box B, \Gamma$.

Finally we consider those cases where the duplicate formula is the main formula in the last step in the proof:

1. The last step is $\frac{\Diamond\Sigma|B,C,B\vee C,\Gamma}{\Diamond\Sigma|B\vee C,B\vee C,\Gamma}(\vee)$: Let d be the depth of the proof of $\Diamond\Sigma \mid B \vee C, B \vee C, \Gamma$. Since $(\vee)$ is strongly invertible we know that $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid B, C, B, C, \Gamma$ with a proof of depth $\leq d-1$. With the induction hypothesis follows $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid C, B, C, \Gamma$ with a proof of depth $\leq d-1$. Again with the induction hypothesis follows $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid B, C, \Gamma$ with a proof of depth $\leq d-1$. Thus $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid B \vee C, \Gamma$ with a proof of depth $\leq d$.
2. The last step is $\frac{\Diamond\Sigma|B,B\wedge C,\Gamma \quad \Diamond\Sigma|C,B\wedge C,\Gamma}{\Diamond\Sigma|B\wedge C,B\wedge C,\Gamma}(\wedge)$, where $A \equiv B \wedge C$: Let d be the depth of the proof of $\Diamond\Sigma \mid B \wedge C, B \wedge C, \Gamma$. Since $(\wedge)$ is strongly invertible we know that $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid B, B, \Gamma$ and $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid C, C, \Gamma$ with proofs of depth $\leq d-1$. With the induction hypothesis follows $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid B, \Gamma$ and $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid C, \Gamma$ with proofs of depth $\leq d-1$. Thus $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid B \wedge C, \Gamma$ with a proof of depth $\leq d$.
3. The last step is $\frac{\Diamond B,\Diamond\Sigma|\Diamond B,\Gamma}{\Diamond\Sigma|\Diamond B,\Diamond B,\Gamma}(\Diamond)$, where $A \equiv \Diamond B$: Let d be the depth of the proof of $\Diamond B, \Diamond\Sigma \mid \Diamond B, \Gamma$. Since $(\Diamond)$ is strongly invertible we know that $\mathsf{K}^{\mathcal{S}} \vdash \Diamond B, \Diamond B, \Diamond\Sigma \mid \Gamma$ with a proof of

depth $\leq d-1$. With the induction hypothesis and an application of $(\Diamond)$ follows $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid \Diamond B, \Gamma$ with a proof of depth $\leq d$.

4. The last step in $\mathcal{P}$ is $\frac{\epsilon \mid B, \Sigma}{\Diamond\Sigma \mid \Box B, \Box B, \Gamma}(\Box)$, where $A \equiv \Box B$: With an application of $(\Box)$ we obtain $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid \Box B, \Gamma$.

5.2.7 THEOREM equivalence of $\mathsf{K}^{\mathcal{G},2}$ and $\mathsf{K}^{\mathcal{S}}$

$$\mathsf{K}^{\mathcal{G},2} \vdash \boxed{\boxed{\epsilon \mid A}} \quad \Leftrightarrow \quad \mathsf{K}^{\mathcal{S}} \vdash \epsilon \mid A$$

Proof

'$\Rightarrow$':

We first explain the idea of the proof. During backward proof search in $\mathsf{K}^{\mathcal{G},2}$ we first add vertices to the K graph (backward applications of $(\Box)$) and later on we jump into one of these vertices (backward application of (jump)), whereas in $\mathsf{K}^{\mathcal{S}}$ we have only one rule $(\Box)$ which does both at once. Therefore we first rearrange the proof in $\mathsf{K}^{\mathcal{G},2}$ such that $(\Box)$ applications occur only right after a $(\Box)$ or a (jump) application. Note that here we make use of the strong invertibility (and not just of the invertibility) of the rules $(\vee)$, $(\wedge)$, $(\Diamond)$. In a second step we remove those backward applications of $(\Box)$ which add vertices to the K graph which are not used. The diagram below illustrates these two steps. Finally we can translate the resulting proof step by step (beginning at the root of the proof) into a proof in $\mathsf{K}^{\mathcal{S}}$ by combining successive (jump) and $(\Box)$ applications into a $(\Box)$ application.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{G_4}\,(\text{jump})}{G_3}\,(\vee)}{G_2}\,(\Box)}{G_1}\,(\Box)}{\vdots}\,(\vee) \quad \rightsquigarrow \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{G_4'}\,(\text{jump})}{G_3'}\,(\Box)}{G_2'}\,(\Box)}{G_1'}\,(\vee)}{\vdots}\,(\vee) \quad \rightsquigarrow \quad \dfrac{\dfrac{\dfrac{\dfrac{\vdots}{G_3''}\,(\text{jump})}{G_2'}\,(\Box)}{G_1'}\,(\vee)}{\vdots}\,(\vee)$$

First, we define a function m on proofs in $\mathsf{K}^{\mathcal{G},2}$. We call an application of $(\Box)$ with consequence G' 'early' if one of the rules $(\vee)$, $(\wedge)$, $(\Diamond)$ is applicable backwards on G'. If $\mathcal{P}$ is a proof without early applications of $(\Box)$, then we set $m(\mathcal{P}) = 0$. Otherwise let $G_1, \ldots, G_k$ be the conclusions of the early applications of $(\Box)$ in $\mathcal{P}$ and for all $i \in \{1, \ldots, n\}$ let n_i be the length of the longest branch in the proof of G_i. We set $m(\mathcal{P}) = \max(\{n_1, \ldots, n_k\})$.

Second, we define a transformation on proofs in $\mathsf{K}^{\mathcal{G},2}$. Let $\mathcal{P}$ be a proof of a K graph G in $\mathsf{K}^{\mathcal{G},2}$ whose last step is an early application of $(\Box)$. Starting with G we apply the rules $(\vee)$, $(\wedge)$, $(\Diamond)$ backwards as long as possible. This process will terminate, and in general we end up with several K graphs $G_1, \ldots, G_n$. Because of theorem 4.2.10 there exist proofs $\mathcal{P}_1, \ldots, \mathcal{P}_n$ of these K graphs. Thus we obtain a proof $\mathcal{P}'$ of G in $\mathsf{K}^{\mathcal{G},2}$ that consists of $\mathcal{P}_1, \ldots, \mathcal{P}_n$ and applications of $(\vee)$, $(\wedge)$, $(\Diamond)$. Because of theorem 4.2.10 we know that the depth of each of $\mathcal{P}_1, \ldots, \mathcal{P}_n$ is at most the depth of $\mathcal{P}$. Since none of $(\vee)$, $(\wedge)$, $(\Box)$ can be applicable backwards on $G_1, \ldots, G_n$, the last step in these proofs must be $(\Box)$ or (jump) application (and if it is a $(\Box)$ application, then it is not an early one). Thus $m(\mathcal{P}') < m(\mathcal{P})$.

Let now $\mathcal{P}$ be a proof of $\boxed{\boxed{\epsilon \mid A}}$ in $\mathsf{K}^{\mathcal{G},2}$. Let $G_1, \ldots, G_k$ be the conclusions of those early applications of $(\Box)$ in $\mathcal{P}$ such that for all $i \in \{1, \ldots, n\}$ there is no early application of $(\Box)$ between G_i and the root of $\mathcal{P}$. Let $\mathcal{P}_1, \ldots, \mathcal{P}_k$ be the proofs of $G_1, \ldots, G_k$. We apply the

transformation described above on these subproofs of $\mathcal{P}$. This process is repeated as long as there are early applications of ($\Box$) left. It terminates because the value computed by the function m always decreases. The result is a proof of $\boxed{\boxed{\epsilon \mid A}}$ in $\mathsf{K}^{\mathcal{G},2}$ without early applications of ($\Box$).
Now we remove those backward applications of ($\Box$) that generate worlds in which we do not jump. The result is still a proof of $\boxed{\boxed{\epsilon \mid A}}$ in $\mathsf{K}^{\mathcal{G},2}$.
This proof can be translated step by step (beginning at the root of the proof) into a proof of $\epsilon \mid A$ in $\mathsf{K}^{\mathcal{S}}$, combining ($\Box$) and (jump) into ($\Box$).
'$\Leftarrow$':
Let $\mathcal{P}$ be a proof of $\epsilon \mid A$ in $\mathsf{K}^{\mathcal{S}}$. We can translate it step by step (beginning at the root) into a proof of $\boxed{\boxed{\epsilon \mid A}}$ in $\mathsf{K}^{\mathcal{G},2}$, replacing ($\Box$) by ($\Box$) and (jump).

5.2.8 THEOREM proofs in $\mathsf{K}^{\mathcal{H}}$ from proofs in $\mathsf{K}^{\mathcal{S}}$

$$\mathsf{K}^{\mathcal{S}} \vdash \epsilon \mid \mathrm{nnf}(A) \quad \Rightarrow \quad \mathsf{K}^{\mathcal{H}} \vdash A$$

Proof

We prove $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\mathrm{nnf}(A_1), \ldots, \Diamond\mathrm{nnf}(A_m) \mid \mathrm{nnf}(B_1), \ldots, \mathrm{nnf}(B_n) \Rightarrow \mathsf{K}^{\mathcal{H}} \vdash \Diamond A_1 \lor \ldots \lor \Diamond A_m \lor B_1 \lor \ldots \lor B_n$ with an induction on proof depth.

1. $\Diamond\mathrm{nnf}(A_1), \ldots, \Diamond\mathrm{nnf}(A_m) \mid \mathrm{nnf}(B_1), \ldots, \mathrm{nnf}(B_n)$ is an instance of the (true) axiom: Then $\Diamond A_1 \lor \ldots \lor \Diamond A_m \lor B_1 \lor \ldots \lor B_n$ is an instance of the (cpc) axiom.
2. $\Diamond\mathrm{nnf}(A_1), \ldots, \Diamond\mathrm{nnf}(A_m) \mid \mathrm{nnf}(B_1), \ldots, \mathrm{nnf}(B_n)$ is an instance of the (id) axiom: Then $\Diamond A_1 \lor \ldots \lor \Diamond A_m \lor B_1 \lor \ldots \lor B_n$ is an instance of the (cpc) axiom.
3. The last step is $\frac{\Diamond\mathrm{nnf}(A_1),\ldots,\Diamond\mathrm{nnf}(A_m)\mid\mathrm{nnf}(C_1),\mathrm{nnf}(C_2),\mathrm{nnf}(B_2),\ldots,\mathrm{nnf}(B_n)}{\Diamond\mathrm{nnf}(A_1),\ldots,\Diamond\mathrm{nnf}(A_m)\mid\mathrm{nnf}(C_1)\lor\mathrm{nnf}(C_2),\mathrm{nnf}(B_2),\ldots,\mathrm{nnf}(B_n)}(\lor)$, where $\mathrm{nnf}(B_1)$ is $\mathrm{nnf}(C_1) \lor \mathrm{nnf}(C_2)$: With the induction hypothesis follows $\mathsf{K}^{\mathcal{H}} \vdash \Diamond A_1 \lor \ldots \lor \Diamond A_m \lor C_1 \lor C_2 \lor B_2 \lor \ldots \lor B_n$, and with (cpc) and (mp) $\mathsf{K}^{\mathcal{H}} \vdash \Diamond A_1 \lor \ldots \lor \Diamond A_m \lor (C_1 \lor C_2) \lor B_2 \lor \ldots \lor B_n$.
4. The last step is
$$\frac{\Diamond\mathrm{nnf}(A_1),\ldots,\Diamond\mathrm{nnf}(A_m)\mid\mathrm{nnf}(C_1),\mathrm{nnf}(B_2),\ldots,\mathrm{nnf}(B_n) \quad \Diamond\mathrm{nnf}(A_1),\ldots,\Diamond\mathrm{nnf}(A_m)\mid\mathrm{nnf}(C_2),\mathrm{nnf}(B_2),\ldots,\mathrm{nnf}(B_n)}{\Diamond\mathrm{nnf}(A_1),\ldots,\Diamond\mathrm{nnf}(A_m)\mid\mathrm{nnf}(C_1)\land\mathrm{nnf}(C_2),\mathrm{nnf}(B_2),\ldots,\mathrm{nnf}(B_n)}(\land)$$
where $\mathrm{nnf}(B_1) \equiv \mathrm{nnf}(C_1) \land \mathrm{nnf}(C_2)$: With the induction hypothesis follows $\mathsf{K}^{\mathcal{H}} \vdash \Diamond A_1 \lor \ldots \lor \Diamond A_m \lor C_1 \lor B_2 \lor \ldots \lor B_n$ and $\mathsf{K}^{\mathcal{H}} \vdash \Diamond A_1 \lor \ldots \lor \Diamond A_m \lor C_2 \lor B_2 \lor \ldots \lor B_n$. With (cpc) and (mp) we obtain $\mathsf{K}^{\mathcal{H}} \vdash \Diamond A_1 \lor \ldots \lor \Diamond A_m \lor (C_1 \land C_2) \lor B_2 \lor \ldots \lor B_n$, as shown in the proof fragment below. (We use the abbreviation $\Diamond D \equiv \Diamond A_1 \lor \ldots \lor \Diamond A_m$).
$$\dfrac{\dfrac{\vdots}{\Diamond D \lor C_2 \lor B_2 \lor \ldots \lor B_n} \qquad \dfrac{\dfrac{\vdots}{\Diamond D \lor C_1 \lor B_2 \lor \ldots \lor B_n} \quad \dfrac{}{\begin{array}{l}(\Diamond D \lor C_1 \lor B_2 \lor \ldots \lor B_n)\\ \to (\Diamond D \lor C_2 \lor B_2 \lor \ldots \lor B_n)\\ \to (\Diamond D \lor (C_1 \land C_2) \lor B_2 \lor \ldots \lor B_n)\end{array}}\text{(cpc)}}{\begin{array}{l}(\Diamond D \lor C_2 \lor B_2 \lor \ldots \lor B_n)\\ \to (\Diamond D \lor (C_1 \land C_2) \lor B_2 \lor \ldots \lor B_n)\end{array}}\text{(mp)}}{\Diamond D \lor (C_1 \land C_2) \lor B_2 \lor \ldots \lor B_n}\text{(mp)}$$
5. The last step is $\frac{\Diamond\mathrm{nnf}(C),\Diamond\mathrm{nnf}(A_1),\ldots,\Diamond\mathrm{nnf}(A_m)\mid\mathrm{nnf}(B_2),\ldots,\mathrm{nnf}(B_n)}{\Diamond\mathrm{nnf}(A_1),\ldots,\Diamond\mathrm{nnf}(A_m)\mid\Diamond\mathrm{nnf}(C),\mathrm{nnf}(B_2),\ldots,\mathrm{nnf}(B_n)}(\Diamond)$, where $\mathrm{nnf}(B_1) \equiv \Diamond\mathrm{nnf}(C)$: With the induction hypothesis we obtain $\mathsf{K}^{\mathcal{H}} \vdash \Diamond C \lor \Diamond A_1 \lor \ldots \lor \Diamond A_m \lor B_2 \lor \ldots \lor B_n$. With (cpc) and (mp) follows $\mathsf{K}^{\mathcal{H}} \vdash \Diamond A_1 \lor \ldots \lor \Diamond A_m \lor \Diamond C \lor B_2 \lor \ldots \lor B_n$.
6. The last step is $\frac{\epsilon\mid\mathrm{nnf}(C),\mathrm{nnf}(A_1),\ldots,\mathrm{nnf}(A_m)}{\Diamond\mathrm{nnf}(A_1),\ldots,\Diamond\mathrm{nnf}(A_m)\mid\Box\mathrm{nnf}(C),\mathrm{nnf}(B_2),\ldots,\mathrm{nnf}(B_n)}(\Box)$, where $\mathrm{nnf}(B_1) \equiv \Box\mathrm{nnf}(C)$: If $m = 0$, then $\mathsf{K}^{\mathcal{H}} \vdash B$ follows with the induction hypothesis, and with

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{C \vee A_1 \vee \ldots \vee A_m}}{\vdots\ (\mathrm{cpc}), (\mathrm{mp})}}{\neg A_1 \to \ldots \to \neg A_m \to C}}{\Box(\neg A_1 \to \ldots \to \neg A_m \to C)}\,(\Box) \quad \dfrac{\vdots}{\begin{array}{l}\Box(\neg A_1 \to \ldots \to \neg A_m \to C)\\ \to \Box\neg A_1 \to \Box(\neg A_2 \to \ldots \to \neg A_m \to C)\end{array}}\,(\mathrm{k})}{\Box\neg A_1 \to \Box(\neg A_2 \to \ldots \to \neg A_m \to C)}\,(\mathrm{mp})}{\vdots\ (\mathrm{k}), (\mathrm{cpc}), (\mathrm{mp})}}{\Box\neg A_1 \to \ldots \to \Box\neg A_m \to \Box C}}{\vdots\ (\Diamond\Box_2), (\mathrm{cpc}), (\mathrm{mp})}}{\dfrac{\Diamond A_1 \vee \ldots \vee \Diamond A_m \vee \Box C}{\dfrac{\vdots\ (\mathrm{cpc}), (\mathrm{mp})}{\Diamond A_1 \vee \ldots \vee \Diamond A_m \vee \Box C \vee B_2 \vee \ldots \vee B_n}}}$$

Figure 5.b: Simulating the ($\Box$) rule of $\mathsf{K}^{\mathcal{S}}$ in $\mathsf{K}^{\mathcal{H}}$.

$$\dfrac{\dfrac{\dfrac{\vdots}{C}}{\Box C}\,(\Box) \qquad \dfrac{}{\Box C \to (\Box C \vee B_2 \vee \ldots \vee B_n)}\,(\mathrm{cpc})}{\Box C \vee B_2 \vee \ldots \vee B_n}\,(\mathrm{mp})$$

we obtain $\mathsf{K}^{\mathcal{H}} \vdash \Box C \vee B_2 \vee \ldots \vee B_n$. Otherwise $m > 0$. With the induction hypothesis follows $\mathsf{K}^{\mathcal{H}} \vdash C \vee A_1 \vee \ldots \vee A_m$. With the proof fragment of figure 5.b we then obtain $\mathsf{K}^{\mathcal{H}} \vdash \Diamond A_1 \vee \ldots \vee \Diamond A_m \vee \Box C \vee B_2 \vee \ldots \vee B_n$.

5.2.9 REMARK simplifying $\mathsf{K}^{\mathcal{S}}$

The only thing we do with $\Diamond$ formulas in $\mathsf{K}^{\mathcal{S}}$ is to put them on the left hand side when applying the ($\Diamond$) rule backwards, and move them back on the right hand side when applying ($\Box$) backwards. Thus we could simplify $\mathsf{K}^{\mathcal{S}}$ to the following calculus:

axioms:

$$\dfrac{}{\mathsf{true}, \Gamma}\,(\mathsf{true}) \qquad\qquad \dfrac{}{P, \neg P, \Gamma}\,(\mathrm{id})$$

rules:

$$\dfrac{A, B, \Gamma}{A \vee B, \Gamma}\,(\vee) \qquad\qquad \dfrac{A, \Gamma \quad B, \Gamma}{A \wedge B, \Gamma}\,(\wedge)$$

$$\dfrac{A, \Sigma}{\Box A, \Diamond\Sigma, \Gamma}\ \text{no } \Diamond \text{ fmls in } \Gamma \quad (\Box)$$

From the point of view of backward proof search, this simplification offers no advantages. (Note that this simplification has considerable disadvantages in the case of KT and $\mathsf{S4}$; see remark 5.4.9.) Therefore we do not consider this calculus in the following, but continue in the same way as in the other sections in this chapter in order to obtain a more uniform presentation.

5.2.10 DEFINITION sequent calculus $\mathsf{K}^{\mathcal{S},2}$

axioms:

$$\frac{}{\Diamond\Sigma \mid \mathsf{true}, \Gamma}\ (\mathsf{true}) \qquad \frac{}{\Diamond\Sigma \mid P, \neg P, \Gamma}\ (\mathrm{id})$$

rules:

$$\frac{\Diamond\Sigma \mid A, B, \Gamma}{\Diamond\Sigma \mid A \lor B, \Gamma}\ (\lor) \qquad \frac{\Diamond\Sigma \mid A, \Gamma \quad \Diamond\Sigma \mid B, \Gamma}{\Diamond\Sigma \mid A \land B, \Gamma}\ (\land)$$

$$\frac{\Diamond A, \Diamond\Sigma \mid \Gamma}{\Diamond\Sigma \mid \Diamond A, \Gamma}\ \Diamond A \notin \Diamond\Sigma \quad (\Diamond, \mathrm{new}) \qquad \frac{\Diamond\Sigma \mid \Gamma}{\Diamond\Sigma \mid \Diamond A, \Gamma}\ \Diamond A \in \Diamond\Sigma \quad (\Diamond, \mathrm{dup})$$

$$\frac{\epsilon \mid A, \Sigma}{\Diamond\Sigma \mid \Box A, \Gamma}\ (\Box)$$

main formulas: true in (true), P and $\neg P$ in (id), $A \lor B$ in $(\lor)$, $A \land B$ in $(\land)$, $\Diamond A$ in $(\Diamond, \mathrm{new})$ and $(\Diamond, \mathrm{dup})$, $\Box A$ in $(\Box)$

side formulas: A and B in $(\lor)$, A and B in $(\land)$, $\Diamond A$ in $(\Diamond, \mathrm{new})$, none in $(\Diamond, \mathrm{dup})$, A in $(\Box)$

Example

$\mathsf{K}^{\mathcal{S},2} \vdash \epsilon \mid \Diamond(p_0 \land \neg p_1) \lor (\Diamond\neg p_0 \lor \Box p_1)$, as the following proof shows (cp. the example in definition 5.2.2).

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\epsilon \mid p_1, p_0, \neg p_0}\ (\mathrm{id}) \quad \dfrac{}{\epsilon \mid p_1, \neg p_1, \neg p_0}\ (\mathrm{id})}{\epsilon \mid p_1, p_0 \land \neg p_1, \neg p_0}\ (\land)}{\Diamond(p_0 \land \neg p_1), \Diamond\neg p_0 \mid \Box p_1}\ (\Box)}{\Diamond(p_0 \land \neg p_1) \mid \Diamond\neg p_0, \Box p_1}\ (\Diamond, \mathrm{new})}{\epsilon \mid \Diamond(p_0 \land \neg p_1), \Diamond\neg p_0, \Box p_1}\ (\Diamond, \mathrm{new})}{\epsilon \mid \Diamond(p_0 \land \neg p_1), \Diamond\neg p_0 \lor \Box p_1}\ (\lor)}{\epsilon \mid \Diamond(p_0 \land \neg p_1) \lor (\Diamond\neg p_0 \lor \Box p_1)}\ (\lor)}{}$$

5.2.11 THEOREM $\mathsf{K}^{\mathcal{S},2}$: weakening

If $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$, then $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$. Moreover, if d is the depth of the proof of $\Diamond\Sigma \mid \Gamma$, then there exists a proof of $\Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$ whose depth is at most d.

Proof

Compared to the corresponding proof for $\mathsf{K}^{\mathcal{S}}$ (theorem 5.2.3), only the cases for $(\Diamond, \mathrm{new})$ and $(\Diamond, \mathrm{dup})$ are different.

1. The last step is $\frac{\Diamond A, \Diamond\Sigma \mid \Gamma}{\Diamond\Sigma \mid \Diamond A, \Gamma}(\Diamond, \mathrm{new})$: If $\Diamond A \notin \Diamond\Delta$, then $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Delta, \Diamond A, \Diamond\Sigma \mid \Pi, \Gamma$ because of the induction hypothesis, and with a $(\Diamond, \mathrm{new})$ application follows $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, \Diamond A, \Gamma$. If $\Diamond A \in \Diamond\Delta$, then $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, \Diamond A, \Gamma$ directly with the induction hypothesis.
2. The last step is $\frac{\Diamond\Sigma \mid \Gamma}{\Diamond\Sigma \mid \Diamond A, \Gamma}(\Diamond, \mathrm{dup})$: With the induction hypothesis we obtain $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, \Diamond A, \Gamma$.

5.2.12 THEOREM $\mathsf{K}^{\mathcal{S},2}$: invertible rules

The rule ($\Box$) of $\mathsf{K}^{\mathcal{S},2}$ is not invertible. All the other rules are strongly invertible.

Proof

We can proceed as in the corresponding proof for $\mathsf{K}^{\mathcal{S}}$ (theorem 5.2.4). We only discuss the cases where ($\Diamond$, new) or ($\Diamond$, dup) is involved, as these are the only rules which are different. With the transformations

$$\dfrac{\dfrac{\dfrac{\vdots}{\Diamond A, \Diamond\Sigma \mid B, C, \Gamma}}{\Diamond A, \Diamond\Sigma \mid B \vee C, \Gamma}\,(\vee)}{\Diamond\Sigma \mid \Diamond A, B \vee C, \Gamma}\,(\Diamond, \text{new}) \quad\rightsquigarrow\quad \dfrac{\dfrac{\dfrac{\vdots}{\Diamond A, \Diamond\Sigma \mid B, C, \Gamma}}{\Diamond\Sigma \mid \Diamond A, B, C, \Gamma}\,(\Diamond, \text{new})}{\Diamond\Sigma \mid \Diamond A, B \vee C, \Gamma}\,(\vee)$$

and

$$\dfrac{\dfrac{\dfrac{\vdots}{\Diamond A, \Diamond\Sigma \mid B, C, \Gamma}}{\Diamond\Sigma \mid \Diamond A, B, C, \Gamma}\,(\Diamond, \text{new})}{\Diamond\Sigma \mid \Diamond A, B \vee C, \Gamma}\,(\vee) \quad\rightsquigarrow\quad \dfrac{\dfrac{\dfrac{\vdots}{\Diamond A, \Diamond\Sigma \mid B, C, \Gamma}}{\Diamond A, \Diamond\Sigma \mid B \vee C, \Gamma}\,(\vee)}{\Diamond\Sigma \mid \Diamond A, B \vee C, \Gamma}\,(\Diamond, \text{new})$$

follows $(\Diamond,\text{new})(\vee)\cdots \rightsquigarrow (\vee)(\Diamond,\text{new})\cdots$ and $(\vee)(\Diamond,\text{new})\cdots \rightsquigarrow (\Diamond,\text{new})(\vee)\cdots$. In the same way we easily obtain $(\Diamond,\text{dup})(\vee)\cdots \rightsquigarrow (\vee)(\Diamond,\text{dup})\cdots$ and $(\vee)(\Diamond,\text{dup})\cdots \rightsquigarrow (\Diamond,\text{dup})(\vee)\cdots$. Also $(\Diamond,\text{new})(\wedge)\cdots \rightsquigarrow (\wedge)(\Diamond,\text{new})\cdots$, $(\wedge)(\Diamond,\text{new})\cdots \rightsquigarrow (\Diamond,\text{new})(\wedge)\cdots$, $(\Diamond,\text{dup})(\wedge)\cdots \rightsquigarrow (\wedge)(\Diamond,\text{dup})\cdots$ and $(\wedge)(\Diamond,\text{dup})\cdots \rightsquigarrow (\Diamond,\text{dup})(\wedge)\cdots$ cause no problems. With the transformation

$$\dfrac{\epsilon \mid A, \Sigma}{\Diamond\Sigma \mid \Box A, \Diamond B, \Gamma}\,(\Box) \quad\rightsquigarrow\quad \dfrac{\dfrac{\epsilon \mid A, B, \Sigma}{\Diamond B, \Diamond\Sigma \mid \Box A, \Gamma}\,(\Box)}{\Diamond\Sigma \mid \Box A, \Diamond B, \Gamma}\,(\Diamond, \text{new})$$

and weakening we obtain $(\Box)\cdots \rightsquigarrow (\Diamond,\text{new})(\Box)\cdots$, and finally we use

$$\dfrac{\dfrac{\vdots}{\epsilon \mid A, \Sigma}}{\Diamond\Sigma \mid \Box A, \Diamond B, \Gamma}\,(\Box) \quad\rightsquigarrow\quad \dfrac{\dfrac{\dfrac{\vdots}{\epsilon \mid A, \Sigma}}{\Diamond\Sigma \mid \Box A, \Gamma}\,(\Box)}{\Diamond\Sigma \mid \Box A, \Diamond B, \Gamma}\,(\Diamond, \text{dup})$$

to show $(\Box)\cdots \rightsquigarrow (\Diamond,\text{dup})(\Box)\cdots$. Note that we cannot switch applications of ($\Diamond$, dup) and ($\Diamond$, new) if they have the same main formula, but this does not hamper their invertibility.

5.2.13 THEOREM $\mathsf{K}^{\mathcal{S},2}$: duplicate formulas

$$\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, A, \Gamma \quad\Rightarrow\quad \mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, \Gamma$$

Proof

We make an induction on proof depth. Simultaneously we prove that the depth of the new proof is at most the depth of the original one. Most cases are the same as in the corresponding proof for $\mathsf{K}^{\mathcal{S}}$ (theorem 5.2.6).

1. $\Diamond\Sigma \mid A, A, \Gamma$ is an instance of the (true) axiom: As in the proof of theorem 5.2.6.
2. $\Diamond\Sigma \mid A, A, \Gamma$ is an instance of the (id) axiom: As in the proof of theorem 5.2.6.
3. The last step is $\frac{\Diamond\Sigma|A,A,B,C,\Gamma}{\Diamond\Sigma|A,A,B\vee C,\Gamma}(\vee)$: As in the proof of theorem 5.2.6.
4. The last step is $\frac{\Diamond\Sigma|A,A,B,\Gamma \quad \Diamond\Sigma|A,A,C,\Gamma}{\Diamond\Sigma|A,A,B\wedge C,\Gamma}(\wedge)$: As in the proof of theorem 5.2.6.
5. The last step is $\frac{\Diamond B,\Diamond\Sigma|A,A,\Gamma}{\Diamond\Sigma|A,A,\Diamond B,\Gamma}(\Diamond, \text{new})$: With the induction hypothesis and an application of $(\Diamond, \text{new})$ we obtain $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, \Diamond B, \Gamma$.
6. The last step is $\frac{\Diamond\Sigma|A,A,\Gamma}{\Diamond\Sigma|A,A,\Diamond B,\Gamma}(\Diamond, \text{dup})$: With the induction hypothesis and an application of $(\Diamond, \text{dup})$ we obtain $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, \Diamond B, \Gamma$.
7. The last step in $\mathcal{P}$ is $\frac{\epsilon|B,\Sigma}{\Diamond\Sigma|A,A,\Box B,\Gamma}(\Box)$: As in the proof of theorem 5.2.6.
8. The last step is $\frac{\Diamond\Sigma|B,C,B\vee C,\Gamma}{\Diamond\Sigma|B\vee C,B\vee C,\Gamma}(\vee)$, where $A \equiv B \vee C$: As in the proof of theorem 5.2.6.
9. The last step is $\frac{\Diamond\Sigma|B,B\wedge C,\Gamma \quad \Diamond\Sigma|C,B\wedge C,\Gamma}{\Diamond\Sigma|B\wedge C,B\wedge C,\Gamma}(\wedge)$, where $A \equiv B \wedge C$: As in the proof of theorem 5.2.6.
10. The last step is $\frac{\Diamond B,\Diamond\Sigma|\Diamond B,\Gamma}{\Diamond\Sigma|\Diamond B,\Diamond B,\Gamma}(\Diamond, \text{new})$, where $A \equiv \Diamond B$: Let d be the depth of the proof of $\Diamond B, \Diamond\Sigma \mid \Diamond B, \Gamma$. Since $(\Diamond, \text{dup})$ is strongly invertible we know that $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond B, \Diamond\Sigma \mid \Gamma$ with a proof of depth $\leq d-1$. With an application of $(\Diamond, \text{new})$ follows $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Diamond B, \Gamma$ with a proof of depth $\leq d$.
11. The last step is $\frac{\Diamond\Sigma|\Diamond B,\Gamma}{\Diamond\Sigma|\Diamond B,\Diamond B,\Gamma}(\Diamond, \text{dup})$, where $A \equiv \Diamond B$: The premise is exactly what we want.
12. The last step in $\mathcal{P}$ is $\frac{\epsilon|B,\Sigma}{\Diamond\Sigma|\Box B,\Box B,\Gamma}(\Box)$, where $A \equiv \Box B$: As in the proof of theorem 5.2.6.

5.2.14 THEOREM equivalence of $\mathsf{K}^{\mathcal{S}}$ and $\mathsf{K}^{\mathcal{S},2}$

$$\mathsf{K}^{\mathcal{S}} \vdash \epsilon \mid A \quad \Leftrightarrow \quad \mathsf{K}^{\mathcal{S},2} \vdash \epsilon \mid A$$

Proof

'$\Rightarrow$':

We prove $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid \Gamma \Rightarrow \mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$ with an induction on proof depth. The only non-trivial case is the translation of a $(\Diamond)$ application.

1. The last step is $\frac{\Diamond A,\Diamond\Sigma|\Gamma}{\Diamond\Sigma|\Diamond A,\Gamma}(\Diamond)$: If $\Diamond A \notin \Diamond\Sigma$, then we have $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond A, \Diamond\Sigma \mid \Gamma$ because of the induction hypothesis, and with a $(\Diamond, \text{new})$ application we obtain $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Diamond A, \Gamma$. If $\Diamond A \in \Diamond\Sigma$, then we have $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid \Gamma$ because of theorem 5.2.6. With the induction hypothesis we obtain $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$, and with a $(\Diamond, \text{dup})$ application follows $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Diamond A, \Gamma$.

'$\Leftarrow$':

We prove $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma \Rightarrow \mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid \Gamma$ with an induction on proof depth. The only non-trivial cases are $(\Diamond, \text{new})$ and $(\Diamond, \text{dup})$ applications.

1. The last step is $\frac{\Diamond A,\Diamond\Sigma|\Gamma}{\Diamond\Sigma|\Diamond A,\Gamma}(\Diamond, \text{new})$: With the induction hypothesis we obtain $\mathsf{K}^{\mathcal{S}} \vdash \Diamond A, \Diamond\Sigma \mid \Gamma$, and with a $(\Diamond)$ application follows $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid \Diamond A, \Gamma$.
2. The last step is $\frac{\Diamond\Sigma|\Gamma}{\Diamond\Sigma|\Diamond A,\Gamma}(\Diamond, \text{dup})$: With the induction hypothesis we obtain $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid \Gamma$, and with weakening follows $\mathsf{K}^{\mathcal{S}} \vdash \Diamond\Sigma \mid \Diamond A, \Gamma$.

5.2.15 THEOREM $\mathsf{K}^{\mathcal{S},2}$: termination

Backward proof search in $\mathsf{K}^{\mathcal{S},2}$ always terminates.

Proof

We define $m(\Diamond\Sigma \mid \Gamma) := \text{length}(\Diamond\Sigma) + \text{length}(\Gamma) - \text{card}(\Diamond\Sigma)$. Obviously always $m(\Diamond\Sigma \mid \Gamma) \geq 0$, and the value computed by m decreases with each backward application of a rule of $\mathsf{K}^{\mathcal{S},2}$:

1. $m(\Diamond\Sigma \mid A \vee B, \Gamma) > m(\Diamond\Sigma \mid A, B, \Gamma)$
2. $m(\Diamond\Sigma \mid A \wedge B, \Gamma) > m(\Diamond\Sigma \mid A, \Gamma)$, $m(\Diamond\Sigma \mid A \wedge B, \Gamma) > m(\Diamond\Sigma \mid B, \Gamma)$
3. $m(\Diamond\Sigma \mid \Diamond A, \Gamma) > m(\Diamond A, \Diamond\Sigma \mid \Gamma)$
4. $m(\Diamond\Sigma \mid \Diamond A, \Gamma) > m(\Diamond\Sigma \mid \Gamma)$
5. $m(\Diamond\Sigma \mid \Box A, \Gamma) > m(\epsilon \mid A, \Sigma)$

5.2.16 THEOREM $\mathsf{K}^{\mathcal{S},2}$: extend (id)

If A is in negation normal form, then:

$$\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, \text{nnf}(\neg A), \Gamma$$

Proof

We make an induction on length(A).

1. $A \equiv \mathsf{true}$ or $A \equiv \mathsf{false}$: Then $\Diamond\Sigma \mid A, \text{nnf}(\neg A), \Gamma$ is an instance of the (true) axiom.
2. A is a variable or a negated variable: Then $\Diamond\Sigma \mid A, \text{nnf}(\neg A), \Gamma$ is an instance of the (id) axiom.
3. $A \equiv B \vee C$: With the induction hypothesis we obtain $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid B, C, \text{nnf}(\neg B), \Gamma$ and $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid B, C, \text{nnf}(\neg C), \Gamma$. With applications of ($\vee$) and ($\wedge$) follows $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid B \vee C, \text{nnf}(\neg B) \wedge \text{nnf}(\neg C), \Gamma$.
4. $A \equiv B \vee C$: Analogous to case 3.
5. $A \equiv \Box B$: If $\Diamond\text{nnf}(\neg B) \notin \Diamond\Sigma$, then we obtain $\mathsf{K}^{\mathcal{S},2} \vdash \epsilon \mid B, \text{nnf}(\neg B), \Sigma$ with the induction hypothesis, and with applications of ($\Box$) and ($\Diamond$, new) follows $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Box B, \Diamond\text{nnf}(\neg B), \Gamma$. If $\Diamond\text{nnf}(\neg B) \in \Diamond\Sigma$, then we obtain $\mathsf{K}^{\mathcal{S},2} \vdash \epsilon \mid B, \Sigma$ with the induction hypothesis, and with applications of ($\Box$) and ($\Diamond$, dup) follows $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Box B, \Diamond\text{nnf}(\neg B), \Gamma$.
6. $A \equiv \Diamond B$: Analogous to case 5.

5.2.17 THEOREM $\mathsf{K}^{\mathcal{S},2}$: use-check

Let $\mathcal{P}$ be a proof in $\mathsf{K}^{\mathcal{S},2}$ of the sequent $\Diamond\Sigma \mid A, \Gamma$. We mark it as follows:

1. The main formulas in the instances of axioms are marked.
2. The main formula in the instances of the ($\Box$) rule is marked.
3. If a side formula is marked, then the main formula is marked.
4. If in the premise of an instance of ($\vee$), ($\wedge$), ($\Diamond$, new) or ($\Diamond$, dup) a formula in Γ is marked, then the corresponding formula in the multiset Γ in the conclusion is marked.
5. If in the premise of an instance of ($\vee$), ($\wedge$), ($\Diamond$, new) or ($\Diamond$, dup) a formula in $\Diamond\Sigma$ is marked, then the corresponding formula in the multiset $\Diamond\Sigma$ in the conclusion is marked.

6. If in the premise of an instance of $(\Box)$ a formula in Σ is marked, then the corresponding formula in the multiset $\Diamond\Sigma$ in the conclusion is marked.

If the formula A in the sequent $\Diamond\Sigma \mid A, \Gamma$ is not marked, then $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$.

Proof

We generalise the theorem: If $\mathcal{P}$ is a marked proof of $\Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$ in $\mathsf{K}^{\mathcal{S},2}$ and no formula in $\Diamond\Delta$ and Π is marked, then $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$. We make an induction on proof depth.

1. $\Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$ is an instance of the (true) axiom: Then $\Diamond\Sigma \mid \Gamma$ is also an instance of the (true) axiom.
2. $\Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$ is an instance of the (id) axiom: Then $\Diamond\Sigma \mid \Gamma$ is also an instance of the (id) axiom.
3. The last step is $\frac{\Diamond\Delta,\Diamond\Sigma|C,D,\Pi,\Gamma}{\Diamond\Delta,\Diamond\Sigma|C\vee D,\Pi,\Gamma}(\vee)$: With the induction hypothesis and an application of $(\vee)$ we obtain $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid C \vee D, \Gamma$.
4. The last step is $\frac{\Diamond\Delta,\Diamond\Sigma|C,\Pi,\Gamma \quad \Diamond\Delta,\Diamond\Sigma|D,\Pi,\Gamma}{\Diamond\Delta,\Diamond\Sigma|C\wedge D,\Pi,\Gamma}(\wedge)$: Analogous to case 3.
5. The last step is $\frac{\Diamond C,\Diamond\Delta,\Diamond\Sigma|\Pi,\Gamma}{\Diamond\Delta,\Diamond\Sigma|\Diamond C,\Pi,\Gamma}(\Diamond,\text{new})$: With the induction hypothesis we obtain $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond C, \Diamond\Sigma \mid \Gamma$, and with an application of $(\Diamond, \text{new})$ follows $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Diamond C, \Gamma$.
6. The last step is $\frac{\Diamond\Delta,\Diamond\Sigma|\Pi,\Gamma}{\Diamond\Delta,\Diamond\Sigma|\Diamond C,\Pi,\Gamma}(\Diamond,\text{dup})$: With the induction hypothesis we obtain $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$, and with an application of $(\Diamond, \text{dup})$ follows $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Diamond C, \Gamma$.
7. The last step is $\frac{\epsilon|C,\Delta,\Sigma}{\Diamond\Delta,\Diamond\Sigma|\Box C,\Pi,\Gamma}(\Box)$: With the induction hypothesis follows $\mathsf{K}^{\mathcal{S},2} \vdash \epsilon \mid C, \Sigma$, and an application of $(\Box)$ gives $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Box C, \Gamma$.
8. The last step is $\frac{\Diamond\Delta,\Diamond\Sigma|C,D,\Pi_1,\Gamma}{\Diamond\Delta,\Diamond\Sigma|C\vee D,\Pi_1,\Gamma}(\vee)$ where $\Pi = C \vee D, \Pi_1$: With the induction hypothesis follows $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$.
9. The last step is $\frac{\Diamond\Delta,\Diamond\Sigma|C,\Pi_1,\Gamma \quad \Diamond\Delta,\Diamond\Sigma|D,\Pi_1,\Gamma}{\Diamond\Delta,\Diamond\Sigma|C\wedge D,\Pi_1,\Gamma}(\vee)$ where $\Pi = C \wedge D, \Pi_1$: With the induction hypothesis follows $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$.
10. The last step is $\frac{\Diamond C,\Diamond\Delta,\Diamond\Sigma|\Pi_1,\Gamma}{\Diamond\Delta,\Diamond\Sigma|\Diamond C,\Pi_1,\Gamma}(\Diamond,\text{new})$ where $\Pi = \Diamond C, \Pi_1$: With the induction hypothesis follows $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$.
11. The last step is $\frac{\Diamond\Delta,\Diamond\Sigma|\Pi_1,\Gamma}{\Diamond\Delta,\Diamond\Sigma|\Diamond C,\Pi_1,\Gamma}(\Diamond,\text{dup})$ where $\Pi = \Diamond C, \Pi_1$: With the induction hypothesis follows $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$.
12. The last step in $\mathcal{P}$ is a $(\Box)$ application with the main formula in Π: This is impossible because all main formulas of $(\Box)$ applications are marked.

Example

We do backward proof search in $\mathsf{K}^{\mathcal{S},2}$ for $\epsilon \mid A$, where $A \equiv ((p_0 \vee p_1) \wedge \Box p_2) \vee \Diamond\neg p_2 \vee (\neg p_0 \wedge \neg p_1)$.

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{(id)}
\UnaryInfC{$\epsilon \mid \mathbf{p_0}, p_1, \Diamond\neg p_2, \neg\mathbf{p_0}$}
\AxiomC{}
\RightLabel{(id)}
\UnaryInfC{$\epsilon \mid p_0, \mathbf{p_1}, \Diamond\neg p_2, \neg\mathbf{p_1}$}
\RightLabel{$(\wedge)$}
\BinaryInfC{$\epsilon \mid \mathbf{p_0}, \mathbf{p_1}, \Diamond\neg p_2, \neg\mathbf{p_0} \wedge \neg\mathbf{p_1}$}
\RightLabel{$(\vee)$}
\UnaryInfC{$\epsilon \mid \mathbf{p_0} \vee \mathbf{p_1}, \Diamond\neg p_2, \neg\mathbf{p_0} \wedge \neg\mathbf{p_1}$}
\AxiomC{}
\RightLabel{(id)}
\UnaryInfC{$\epsilon \mid \mathbf{p_2}, \neg\mathbf{p_2}$}
\RightLabel{$(\Box)$}
\UnaryInfC{$\Diamond\neg\mathbf{p_2} \mid \Box\mathbf{p_2}, \neg p_0$}
\RightLabel{$(\Diamond, \text{new})$}
\UnaryInfC{$\epsilon \mid \Box\mathbf{p_2}, \Diamond\neg\mathbf{p_2}, \neg p_0$}
\AxiomC{\textit{cut off}}
\RightLabel{$(\wedge)$}
\BinaryInfC{$\epsilon \mid \Box\mathbf{p_2}, \Diamond\neg\mathbf{p_2}, \neg p_0 \wedge \neg p_1$}
\RightLabel{$(\wedge)$}
\BinaryInfC{$\epsilon \mid (\mathbf{p_0} \vee \mathbf{p_1}) \wedge \Box\mathbf{p_2}, \Diamond\neg\mathbf{p_2}, \neg\mathbf{p_0} \wedge \neg\mathbf{p_1}$}
\RightLabel{$(\vee)$}
\UnaryInfC{$\epsilon \mid ((\mathbf{p_0} \vee \mathbf{p_1}) \wedge \Box\mathbf{p_2}) \vee \Diamond\neg\mathbf{p_2}, \neg\mathbf{p_0} \wedge \neg\mathbf{p_1}$}
\RightLabel{$(\vee)$}
\UnaryInfC{$\epsilon \mid ((\mathbf{p_0} \vee \mathbf{p_1}) \wedge \Box\mathbf{p_2}) \vee \Diamond\neg\mathbf{p_2} \vee (\neg\mathbf{p_0} \wedge \neg\mathbf{p_1})$}
\end{prooftree}
$$

Note that we can cut off a branch although $\neg p_0 \wedge \neg p_1$ is not a superfluous subformula of A; it is only superfluous on the right branch. Here lemma generation (cp. remark 5.1.21) would not help. We would obtain the following proof (using a rule (dup) to remove duplicate formulas):

$$
\begin{prooftree}
\AxiomC{$\vdots$}
\RightLabel{$(\vee)$}
\UnaryInfC{$\epsilon \mid p_0 \vee p_1, \Diamond\neg p_2, \neg p_0 \wedge \neg p_1$}
\AxiomC{}
\RightLabel{(id)}
\UnaryInfC{$\epsilon \mid p_2, \neg p_2$}
\RightLabel{$(\Box)$}
\UnaryInfC{$\Diamond\neg p_2 \mid \Box p_2, \neg p_0$}
\RightLabel{$(\Diamond, \text{new})$}
\UnaryInfC{$\epsilon \mid \Box p_2, \Diamond\neg p_2, \neg p_0$}
\AxiomC{}
\RightLabel{(id)}
\UnaryInfC{$\epsilon \mid p_2, \neg p_2$}
\RightLabel{$(\Box)$}
\UnaryInfC{$\Diamond\neg p_2 \mid \Box p_2, p_0, \neg p_1$}
\RightLabel{$(\Diamond, \text{new})$}
\UnaryInfC{$\epsilon \mid \Box p_2, \Diamond\neg p_2, p_0, \neg p_1$}
\RightLabel{$(\wedge')$}
\BinaryInfC{$\epsilon \mid \Box p_2, \Diamond\neg p_2, \neg p_0 \wedge \neg p_1$}
\RightLabel{(dup)}
\UnaryInfC{$\epsilon \mid \Box p_2, \Diamond\neg p_2, \neg p_0 \wedge \neg p_1, \neg p_0 \wedge \neg p_1$}
\RightLabel{$(\wedge')$}
\BinaryInfC{$\epsilon \mid (p_0 \vee p_1) \wedge \Box p_2, \Diamond\neg p_2, \neg p_0 \wedge \neg p_1$}
\RightLabel{$(\vee)$}
\UnaryInfC{$\epsilon \mid ((p_0 \vee p_1) \wedge \Box p_2) \vee \Diamond\neg p_2, \neg p_0 \wedge \neg p_1$}
\RightLabel{$(\vee)$}
\UnaryInfC{$\epsilon \mid ((p_0 \vee p_1) \wedge \Box p_2) \vee \Diamond\neg p_2 \vee (\neg p_0 \wedge \neg p_1)$}
\end{prooftree}
$$

5.2.18 REMARK use-check: $\mathsf{K}^{\mathcal{S},2}$ vs. CPC

We cannot use the same formulation as for CPC, because it is possible that a formula is not used in an axiom, but helps to 'go on' with backward proof search. The 'reason' for this behaviour is the non-reflexivity of the accessibility relation in the possible world semantics for K.

Example

We have $\mathsf{K}^{\mathcal{S},2} \vdash \epsilon \mid \Box p_1, \Diamond(p_0 \vee \neg p_0)$, and the only axiom of the proof is $\epsilon \mid p_1, p_0, \neg p_0$, i.e. no subformula of $\Box p_1$ occurs as a main formula in an axiom of the proof.

However $\mathsf{K}^{\mathcal{S},2} \nvdash \epsilon \mid \Box p_1 \wedge p_1, \Diamond(p_0 \vee \neg p_0)$, because on the right branch the sequent $\epsilon \mid p_1, \Diamond(p_0 \vee \neg p_0)$ is not provable (no rule is applicable backwards). Indeed with the use-check for $\mathsf{K}^{\mathcal{S},2}$ we are not allowed to cut off the right branch since $\Box p_1$ has been used on the left branch.

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{id}
\UnaryInfC{$\epsilon \mid p_1, \mathbf{p_0}, \neg\mathbf{p_0}$}
\RightLabel{$(\vee)$}
\UnaryInfC{$\epsilon \mid p_1, \mathbf{p_0} \vee \neg\mathbf{p_0}$}
\RightLabel{$(\Box)$}
\UnaryInfC{$\Diamond(\mathbf{p_0} \vee \neg\mathbf{p_0}) \mid \Box\mathbf{p_1}$}
\RightLabel{$(\Diamond, \text{new})$}
\UnaryInfC{$\epsilon \mid \Box\mathbf{p_1}, \Diamond(\mathbf{p_0} \vee \neg\mathbf{p_0})$}
\AxiomC{$\epsilon \mid p_1, \Diamond(p_0 \vee \neg p_0)$}
\RightLabel{$(\wedge)$}
\BinaryInfC{$\epsilon \mid \Box\mathbf{p_1} \wedge \mathbf{p_1}, \Diamond(\mathbf{p_0} \vee \neg\mathbf{p_0})$}
\RightLabel{$(\vee)$}
\UnaryInfC{$\epsilon \mid (\Box\mathbf{p_1} \wedge \mathbf{p_1}) \vee \Diamond(\mathbf{p_0} \vee \neg\mathbf{p_0})$}
\end{prooftree}
$$

5.2.19 REMARK $\mathsf{K}^{\mathcal{S},2}$: backward proof search

In contrast to backward proof search in $\mathsf{K}^{\mathcal{G},2}$, we deal with sequents and not with trees of sequents. In order to construct a countermodel from a failed backward proof search, we have to gather the literals we put into the worlds of $\mathcal{M}$ from the corresponding sequents.

Example

We do backward proof search in $\mathsf{K}^{\mathcal{S},2}$ for $\epsilon \mid (\Diamond(\neg p_0 \vee p_3) \vee \Diamond\neg p_1) \vee (\Box p_2 \vee \Box(p_0 \wedge p_1))$ (as in the first example in 4.2.12). See figure 5.c for the search tree. The non-proof that corresponds to the failing branch of the search tree look as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\epsilon \mid p_2, \neg p_0, p_3, \neg p_1}{\epsilon \mid p_2, \neg p_0 \lor p_3, \neg p_1}\,(\lor)}{\Diamond(\neg p_0 \lor p_3), \Diamond\neg p_1 \mid \Box p_2, \Box(p_0 \land p_1)}\,(\Box)}{\Diamond(\neg p_0 \lor p_3) \mid \Diamond\neg p_1, \Box p_2, \Box(p_0 \land p_1)}\,(\Diamond, \text{new})}{\epsilon \mid \Diamond(\neg p_0 \lor p_3), \Diamond\neg p_1, \Box p_2, \Box(p_0 \land p_1)}\,(\Diamond, \text{new})}{\epsilon \mid \Diamond(\neg p_0 \lor p_3), \Diamond\neg p_1, \Box p_2 \lor \Box(p_0 \land p_1)}\,(\lor)}{\epsilon \mid \Diamond(\neg p_0 \lor p_3) \lor \Diamond\neg p_1, \Box p_2 \lor \Box(p_0 \land p_1)}\,(\lor)}{\epsilon \mid (\Diamond(\neg p_0 \lor p_3) \lor \Diamond\neg p_1) \lor (\Box p_2 \lor \Box(p_0 \land p_1))}\,(\lor)$$

The proof we find on the other branch of the search tree:

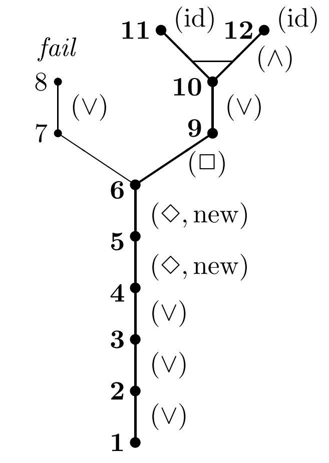
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\epsilon \mid p_0, \neg p_0, p_3, \neg p_1}\,(\text{id}) \qquad \dfrac{}{\epsilon \mid p_1, \neg p_1, p_3, \neg p_1}\,(\text{id})}{\epsilon \mid p_0 \land p_1, \neg p_0, p_3, \neg p_1}\,(\land)}{\epsilon \mid p_0 \land p_1, \neg p_0 \lor p_3, \neg p_1}\,(\lor)}{\Diamond(\neg p_0 \lor p_3), \Diamond\neg p_1 \mid \Box p_2, \Box(p_0 \land p_1)}\,(\Box)}{\Diamond(\neg p_0 \lor p_3) \mid \Diamond\neg p_1, \Box p_2, \Box(p_0 \land p_1)}\,(\Diamond, \text{new})}{\epsilon \mid \Diamond(\neg p_0 \lor p_3), \Diamond\neg p_1, \Box p_2, \Box(p_0 \land p_1)}\,(\Diamond, \text{new})}{\epsilon \mid \Diamond(\neg p_0 \lor p_3), \Diamond\neg p_1, \Box p_2 \lor \Box(p_0 \land p_1)}\,(\lor)}{\epsilon \mid \Diamond(\neg p_0 \lor p_3) \lor \Diamond\neg p_1, \Box p_2 \lor \Box(p_0 \land p_1)}\,(\lor)}{\epsilon \mid (\Diamond(\neg p_0 \lor p_3) \lor \Diamond\neg p_1) \lor (\Box p_2 \lor \Box(p_0 \land p_1))}\,(\lor)$$

Now we do backward proof search for $\epsilon \mid (\Diamond p_3 \lor \Diamond\neg p_1) \lor (\Box(p_2 \lor p_0) \lor \Box(p_0 \land p_1))$ (as in the second example in 4.2.12). See figure 5.d for the search tree. The non-proof that corresponds to the left branch of the search tree look as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\epsilon \mid p_2, p_0, p_3, \neg p_1}{\epsilon \mid p_2 \lor p_0, p_3, \neg p_1}\,(\lor)}{\Diamond p_3, \Diamond\neg p_1 \mid \Box(p_2 \lor p_0), \Box(p_0 \land p_1)}\,(\Box)}{\Diamond p_3 \mid \Diamond\neg p_1, \Box(p_2 \lor p_0), \Box(p_0 \land p_1)}\,(\Diamond, \text{new})}{\epsilon \mid \Diamond p_3, \Diamond\neg p_1, \Box(p_2 \lor p_0), \Box(p_0 \land p_1)}\,(\Diamond, \text{new})}{\epsilon \mid \Diamond p_3, \Diamond\neg p_1, \Box(p_2 \lor p_0) \lor \Box(p_0 \land p_1)}\,(\lor)}{\epsilon \mid \Diamond p_3 \lor \Diamond\neg p_1, \Box(p_2 \lor p_0) \lor \Box(p_0 \land p_1)}\,(\lor)}{\epsilon \mid (\Diamond p_3 \lor \Diamond\neg p_1) \lor (\Box(p_2 \lor p_0) \lor \Box(p_0 \land p_1))}\,(\lor)$$

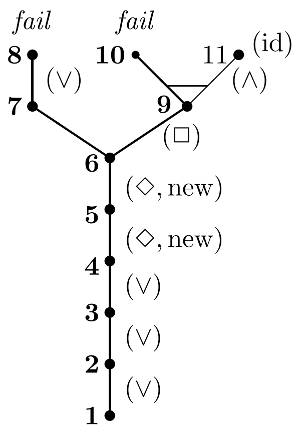
Also the second attempt fails, giving us the following non-proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\epsilon \mid p_0, p_3, \neg p_1 \qquad \dfrac{}{\epsilon \mid p_1, p_3, \neg p_1}\,(\text{id})}{\epsilon \mid p_0 \land p_1, p_3, \neg p_1}\,(\land)}{\Diamond p_3, \Diamond\neg p_1 \mid \Box(p_2 \lor p_0), \Box(p_0 \land p_1)}\,(\Box)}{\Diamond p_3 \mid \Diamond\neg p_1, \Box(p_2 \lor p_0), \Box(p_0 \land p_1)}\,(\Diamond, \text{new})}{\epsilon \mid \Diamond p_3, \Diamond\neg p_1, \Box(p_2 \lor p_0), \Box(p_0 \land p_1)}\,(\Diamond, \text{new})}{\epsilon \mid \Diamond p_3, \Diamond\neg p_1, \Box(p_2 \lor p_0) \lor \Box(p_0 \land p_1)}\,(\lor)}{\epsilon \mid \Diamond p_3 \lor \Diamond\neg p_1, \Box(p_2 \lor p_0) \lor \Box(p_0 \land p_1)}\,(\lor)}{\epsilon \mid (\Diamond p_3 \lor \Diamond\neg p_1) \lor (\Box(p_2 \lor p_0) \lor \Box(p_0 \land p_1))}\,(\lor)$$



The search starts at node 1. After backward applications of (∨) and (◇, new) we have two possibilities to apply (□) backwards (node 6). The first fails (node 8). The second one leads to a branching caused by (∧) and ends in two instances of the (id) axiom (nodes 11 and 12).
Thus $\mathsf{K}^{\mathcal{S},2} \vdash \epsilon \mid (\Diamond(\neg p_0 \vee p_3) \vee \Diamond\neg p_1) \vee (\Box p_2 \vee \Box(p_0 \wedge p_1))$. The thick lines correspond to the proof.

Figure 5.c: The search tree of the first example in remark 5.2.19.



The search starts at node 1. After backward applications of (∨) and (◇, new) we have two possibilities to apply (□) backwards (node 6). The first fails (node 8). The second one leads to a branching caused by (∧). One of these subbranches ends in an instance of the (id) axiom (node 11), the other one fails (node 10).
Thus $\mathsf{K}^{\mathcal{S},2} \nvdash \epsilon \mid (\Diamond p_3 \vee \Diamond\neg p_1) \vee (\Box(p_2 \vee p_0) \vee \Box(p_0 \wedge p_1))$. The thick lines correspond to the countermodel.

Figure 5.d: The search tree of the second example in remark 5.2.19.

The countermodel we can extract is the one we already obtained when searching in $\mathsf{K}^{\mathcal{G},2}$ (see figure 4.p).

5.3 $\mathsf{K} + T$

5.3.1 DEFINITION sequent calculus $(\mathsf{K} + T)^{\mathcal{S}}$

$(\mathsf{K} + T)^{\mathcal{S}}$ is the calculus $\mathsf{K}^{\mathcal{S}}$ with the (□) rule replaced by the rule

$$\frac{\epsilon \mid A, \Sigma, \mathrm{nnf}(\neg T)}{\Diamond\Sigma \mid \Box A, \Gamma}\ (\Box)$$

The main formula of this rule is $\Box A$.

5.3.2 THEOREM $(\mathsf{K}+T)^{\mathcal{S}}$: weakening

If $(\mathsf{K}+T)^{\mathcal{S}} \vdash \Diamond\Sigma \mid \Gamma$, then $(\mathsf{K}+T)^{\mathcal{S}} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$. Moreover, if d is the depth of the proof of $\Diamond\Sigma \mid \Gamma$, then there exists a proof of $\Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$ whose depth is at most d.

Proof

We make an induction on proof depth as in the corresponding proof for $\mathsf{K}^{\mathcal{S}}$ (theorem 5.2.3). The new $(\Box)$ rule causes no problems.

5.3.3 THEOREM $(\mathsf{K}+T)^{\mathcal{S}}$: invertible rules

The rule $(\Box)$ of $(\mathsf{K}+T)^{\mathcal{S}}$ is not invertible. All the other rules are strongly invertible.

Proof

Analogous to the corresponding proof for $\mathsf{K}^{\mathcal{S}}$ (theorem 5.2.4). The new $(\Box)$ rule causes no problems.

5.3.4 THEOREM $(\mathsf{K}+T)^{\mathcal{S}}$: duplicate formulas

$$\begin{array}{rcl}(\mathsf{K}+T)^{\mathcal{S}} \vdash \Diamond\Sigma \mid A, A, \Gamma & \Rightarrow & (\mathsf{K}+T)^{\mathcal{S}} \vdash \Diamond\Sigma \mid A, \Gamma \\ (\mathsf{K}+T)^{\mathcal{S}} \vdash \Diamond A, \Diamond A, \Diamond\Sigma \mid \Gamma & \Rightarrow & (\mathsf{K}+T)^{\mathcal{S}} \vdash \Diamond A, \Diamond\Sigma \mid \Gamma\end{array}$$

Proof

The only difference between this proof and the corresponding proof for $\mathsf{K}^{\mathcal{S}}$ (theorem 5.2.6) is the multiset $\mathrm{nnf}(\neg T)$ in the premises of $(\Box)$ application.

5.3.5 THEOREM equivalence of $(\mathsf{K}+T)^{\mathcal{G},2}$ and $(\mathsf{K}+T)^{\mathcal{S}}$

$$(\mathsf{K}+T)^{\mathcal{G},2} \vdash \boxed{\boxed{\epsilon \mid A, \mathrm{nnf}(\neg T)}} \quad \Leftrightarrow \quad (\mathsf{K}+T)^{\mathcal{S}} \vdash \epsilon \mid A, \mathrm{nnf}(\neg T)$$

Proof

The only difference between this proof and the corresponding proof for $\mathsf{K}^{\mathcal{G},2}$ and $\mathsf{K}^{\mathcal{S}}$ (theorem 5.2.7) is the multiset $\mathrm{nnf}(\neg T)$ in the premises of $(\Box)$ applications.

5.3.6 THEOREM proofs in $(\mathsf{K}+T)^{\mathcal{H}}$ from proofs in $(\mathsf{K}+T)^{\mathcal{S}}$

$$(\mathsf{K}+T)^{\mathcal{S}} \vdash \epsilon \mid \mathrm{nnf}(A), \mathrm{nnf}(\neg T) \quad \Rightarrow \quad (\mathsf{K}+T)^{\mathcal{H}} \vdash A$$

Proof

We can reuse most parts of the proof of theorem 5.2.8. In addition, we have to show that if $T = C_1, \ldots, C_n$, then $(\mathsf{K}+T)^{\mathcal{H}} \vdash B \vee C_1 \vee \ldots \vee C_n$ implies $(\mathsf{K}+T)^{\mathcal{H}} \vdash B$. This follows from $(\mathsf{K}+T)^{\mathcal{S}} \vdash C_i$ (for all $i \in \{1, \ldots, n\}$) with applications of (cpc) and (mp).

5.3.7 DEFINITION sequent calculus $(\mathsf{K}+T)^{\mathcal{S},2}$

$(\mathsf{K}+T)^{\mathcal{S},2}$ is the calculus $\mathsf{K}^{\mathcal{S},2}$ with the $(\Box)$ rule replaced by the rule

$$\frac{\epsilon \mid A, \Sigma, \mathrm{nnf}(\neg T)}{\Diamond\Sigma \mid \Box A, \Gamma}\ (\Box)$$

The main formula of this rule is $\Box A$.

5.3.8 THEOREM $(\mathsf{K}+T)^{\mathcal{S},2}$: weakening

If $(\mathsf{K}+T)^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$, then $(\mathsf{K}+T)^{\mathcal{S},2} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$. Moreover, if d is the depth of the proof of $\Diamond\Sigma \mid \Gamma$, then there exists a proof of $\Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$ whose depth is at most d.

Proof

We make an induction on proof depth as in the corresponding proof for $\mathsf{K}^{\mathcal{S},2}$ (theorem 5.2.11). The different $(\Box)$ rule causes no problems.

5.3.9 THEOREM $(\mathsf{K}+T)^{\mathcal{S},2}$: invertible rules

The rule $(\Box)$ of $(\mathsf{K}+T)^{\mathcal{S},2}$ is not invertible. All the other rules are strongly invertible.

Proof

Analogous to the corresponding proof for $\mathsf{K}^{\mathcal{S},2}$ (theorem 5.2.12). The different $(\Box)$ rule causes no problems.

5.3.10 THEOREM $(\mathsf{K}+T)^{\mathcal{S},2}$: duplicate formulas

$$(\mathsf{K}+T)^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, A, \Gamma \quad \Rightarrow \quad (\mathsf{K}+T)^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, \Gamma$$

Proof

The only difference between this proof and the corresponding proof for $\mathsf{K}^{\mathcal{S},2}$ (theorem 5.2.13) is the multiset $\mathrm{nnf}(\neg T)$ in the premises of $(\Box)$ applications.

5.3.11 THEOREM equivalence of $(\mathsf{K}+T)^{\mathcal{S}}$ and $(\mathsf{K}+T)^{\mathcal{S},2}$

$$(\mathsf{K}+T)^{\mathcal{S}} \vdash \epsilon \mid A, \mathrm{nnf}(\neg T) \quad \Leftrightarrow \quad (\mathsf{K}+T)^{\mathcal{S},2} \vdash \epsilon \mid A, \mathrm{nnf}(\neg T)$$

Proof

The difference between $(\mathsf{K}+T)^{\mathcal{S}}$ and $(\mathsf{K}+T)^{\mathcal{S},2}$ is the same as between $\mathsf{K}^{\mathcal{S}}$ and $\mathsf{K}^{\mathcal{S},2}$. We can use the same proof as for theorem 5.2.14.

5.3.12 REMARK $(\mathsf{K}+T)^{\mathcal{S},2}$: non-termination

In general, backward proof search in $(\mathsf{K}+T)^{\mathcal{S},2}$ does not terminate. With a depth-first strategy the search does not even terminate for provable formulas.

Example

If $T = \Diamond p_0 \vee \Diamond p_1$, then backward proof search for $\Diamond p_0 \vee p_1, \mathrm{nnf}(\neg T)$ in $(\mathsf{K}+T])^{\mathcal{S},2}$ does not terminate, whatever search strategy we use.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\epsilon \mid \neg p_0, p_0, \Box\neg p_0 \wedge \Box\neg p_1}\,(\mathrm{id})}{\Diamond p_0 \mid p_1, \Box\neg p_0}\,(\Box)}{\epsilon \mid \Diamond p_0, p_1, \Box\neg p_0}\,(\Diamond, \mathrm{new}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\epsilon \mid \neg p_0, \Box\neg p_0 \wedge \Box\neg p_1}}{\epsilon \mid \neg p_0, \Box\neg p_0}\,(\Box) \quad \vdots}{\epsilon \mid \neg p_0, \Box\neg p_0 \wedge \Box\neg p_1}\,(\wedge)}{\epsilon \mid \neg p_1, p_0, \Box\neg p_0}\,(\Box) \qquad \dfrac{\vdots}{\epsilon \mid \neg p_1, p_0, \Box\neg p_1}\,(\Box)}{\epsilon \mid \neg p_1, p_0, \Box\neg p_0 \wedge \Box\neg p_1}\,(\wedge)}{\Diamond p_0 \mid p_1, \Box\neg p_1}\,(\Box)}{\epsilon \mid \Diamond p_0, p_1, \Box\neg p_1}\,(\Diamond, \mathrm{new})}{\dfrac{\epsilon \mid \Diamond p_0, p_1, \Box\neg p_0 \wedge \Box\neg p_1}{\epsilon \mid \Diamond p_0 \vee p_1, \Box\neg p_0 \wedge \Box\neg p_1}\,(\vee)}\,(\wedge)$$

5.3.13 REMARK from $(\mathsf{K}+T)^{\mathcal{S},2}$ to $(\mathsf{K}+T)^{\mathcal{S},3}$

We add a history to the sequents of $(\mathsf{K}+T)^{\mathcal{S},2}$. It is sufficient to do the loop-check in $(\Box)$ and to put the $\Box$ and $\Diamond$ formulas of this step into the history. The history is thus a multiset of multisets of formulas. We forbid duplicate $\Diamond$ formulas in Σ and therefore there is only a finite number possible elements of the history during backward proof search. Thus the backward proof search terminates (cp. theorem 5.3.16).

5.3.14 DEFINITION sequent calculus $(\mathsf{K}+T)^{\mathcal{S},3}$

axioms:

$$\dfrac{}{H \mid \Diamond\Sigma \mid \mathsf{true}, \Gamma}\;(\mathsf{true}) \qquad\qquad \dfrac{}{H \mid \Diamond\Sigma \mid P, \neg P, \Gamma}\;(\mathrm{id})$$

rules:

$$\dfrac{H \mid \Diamond\Sigma \mid A, B, \Gamma}{H \mid \Diamond\Sigma \mid A \vee B, \Gamma}\;(\vee) \qquad\qquad \dfrac{H \mid \Diamond\Sigma \mid A, \Gamma \quad H \mid \Diamond\Sigma \mid B, \Gamma}{H \mid \Diamond\Sigma \mid A \wedge B, \Gamma}\;(\wedge)$$

$$\dfrac{H \mid \Diamond A, \Diamond\Sigma \mid \Gamma}{H \mid \Diamond\Sigma \mid \Diamond A, \Gamma}\;\Diamond A \notin \Diamond\Sigma \quad (\Diamond, \mathrm{new}) \qquad\qquad \dfrac{H \mid \Diamond\Sigma \mid \Gamma}{H \mid \Diamond\Sigma \mid \Diamond A, \Gamma}\;\Diamond A \in \Diamond\Sigma \quad (\Diamond, \mathrm{dup})$$

$$\dfrac{[\Box A, \Diamond\Sigma], H \mid \epsilon \mid A, \Sigma, \mathrm{nnf}(\neg T)}{H \mid \Diamond\Sigma \mid \Box A, \Gamma}\;[\Box A, \Diamond\Sigma] \notin H \quad (\Box)$$

main formulas: true in (true), P and $\neg P$ in (id), $A \vee B$ in $(\vee)$, $A \wedge B$ in $(\wedge)$, $\Diamond A$ in $(\Diamond, \mathrm{new})$ and $(\Diamond, \mathrm{dup})$, $\Box A$ in $(\Box)$

side formulas: A and B in $(\vee)$, A and B in $(\wedge)$, $\Diamond A$ in $(\Diamond, \mathrm{new})$, none in $(\Diamond, \mathrm{dup})$, A in $(\Box)$

5.3.15 THEOREM equivalence of $(\mathsf{K}+T)^{\mathcal{S},2}$ and $(\mathsf{K}+T)^{\mathcal{S},3}$

$$(\mathsf{K}+T)^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma, \mathrm{nnf}(\neg T) \quad \Leftrightarrow \quad (\mathsf{K}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma, \mathrm{nnf}(\neg T)$$

Proof

'$\Leftarrow$':

In order to obtain a proof of $\Diamond\Sigma \mid \Gamma$ in $(\mathsf{K}+T)^{\mathcal{S},2}$ from a proof of $\epsilon \mid \Diamond\Sigma \mid \Gamma$ in $(\mathsf{K}+T)^{\mathcal{S},3}$ we simply omit the history. We make an induction on proof depth. The only non-trivial case is an application of the $(\Box)$ rule.

Assume that the last step is $\frac{[\Box A,\Diamond\Sigma],H|\epsilon|A,\Sigma,\mathrm{nnf}(\neg T)}{H|\Diamond\Sigma|\Box A,\Gamma}(\Box)$. Then $\frac{\epsilon|A,\Sigma,\mathrm{nnf}(\neg T)}{\Diamond\Sigma|\Box A,\Gamma}$ is an instance of the $(\Box)$ rule of $(\mathsf{K}+T)^{\mathcal{S}}$. The condition '$[\Box A,\Diamond\Sigma] \notin H$' of the $(\Box)$ rule $(\mathsf{K}+T)^{\mathcal{S},3}$ has no effect.

'$\Rightarrow$':

We prove $(\mathsf{K}+T)^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma \Rightarrow (\mathsf{K}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma$. The basic idea is to take the proof in $(\mathsf{K}+T)^{\mathcal{S},2}$ and to add (beginning at the root) the appropriate history. However, in general the result is not a proof in $(\mathsf{K}+T)^{\mathcal{S},3}$, as the condition '$[\Box A,\Diamond\Sigma] \notin H$' of the $(\Box)$ rule is not satisfied. This condition is the loop-check, i.e. $[\Box A,\Diamond\Sigma] \in H$ means that there is a sequent that occurs twice on a branch. Therefore we first have to remove these superfluous parts from the proof in $(\mathsf{K}+T)^{\mathcal{S},2}$ and add the history only afterwards.

We first define a transformation from proofs of a sequent $\Diamond\Sigma \mid \Gamma$ in $(\mathsf{K}+T)^{\mathcal{S},2}$ into proofs of $\Diamond\Sigma \mid \Gamma$ in $(\mathsf{K}+T)^{\mathcal{S},2}$ which reduces the number of nodes in the proof. We say that a proof in $(\mathsf{K}+T)^{\mathcal{S},2}$ has repetitions if there is a branch on which the same sequent occurs twice. Let $\mathcal{P}$ be a proof of Γ in $(\mathsf{K}+T)^{\mathcal{S},2}$ with repetitions, and let S be the sequent that occurs twice. If we remove the part between these two S together with one occurrence of S, then we obtain a proof of $\Diamond\Sigma \mid \Gamma$ in $(\mathsf{K}+T)^{\mathcal{S},2}$ with fewer nodes. See figure 5.e for an illustration of this transformation in the important case where S is both times the premise of a $(\Box)$ application.

Now we consider the proof of $\Diamond\Sigma \mid \Gamma$ in $(\mathsf{K}+T)^{\mathcal{S},2}$ and apply the transformation described above as long as possible. As the number of nodes in the proof decreases, this process will terminate. The result is still a proof of $\Diamond\Sigma \mid \Gamma$ in $(\mathsf{K}+T)^{\mathcal{S},2}$.

Afterwards we add histories to each sequent, beginning at the root. The result is a proof of $\epsilon \mid \Diamond\Sigma \mid \Gamma$ in $(\mathsf{K}+T)^{\mathcal{S},3}$.

1. The history is empty at the beginning, i.e. the root $\Diamond\Sigma \mid \Gamma$ is transformed into $\epsilon \mid \Diamond\Sigma \mid \Gamma$.
2. $\frac{\Diamond\Sigma|A,B,\Gamma}{\Diamond\Sigma|A\lor B,\Gamma}(\lor)$ is a step in the proof, and the sequent $\Diamond\Sigma \mid A\lor B,\Gamma$ was transformed into $H \mid \Diamond\Sigma \mid A\lor B,\Gamma$: Then $\Diamond\Sigma \mid A,B,\Gamma$ is transformed into $H \mid \Diamond\Sigma \mid A,B,\Gamma$.
3. $\frac{\Diamond\Sigma|A,\Gamma \quad B,\Gamma}{\Diamond\Sigma|A\land B,\Gamma}(\land)$ is a step in the proof, and the sequent $\Diamond\Sigma \mid A\land B,\Gamma$ was transformed into $H \mid \Diamond\Sigma \mid A\land B,\Gamma$: Then $\Diamond\Sigma \mid A,\Gamma$ is transformed into $H \mid \Diamond\Sigma \mid A,\Gamma$ and $\Diamond\Sigma \mid B,\Gamma$ is transformed into $H \mid \Diamond\Sigma \mid B,\Gamma$.
4. $\frac{\epsilon|A,\Sigma,\mathrm{nnf}(\neg T}{\Diamond\Sigma|\Box A,\Gamma}(\Box)$ is a step in the proof, and the sequent $\Diamond\Sigma \mid \Box A,\Gamma$ was transformed into $H \mid \Diamond\Sigma \mid \Box A,\Gamma$: Then $\epsilon \mid A,\Sigma,\mathrm{nnf}(\neg T)$ is transformed into $[\Box A,\Diamond\Sigma],H \mid \epsilon \mid A,\Sigma,\mathrm{nnf}(\neg T)$. Because of the transformation we know that between the root of the new proof and this $(\Box)$ application there is no $(\Box)$ application with a premise of the form $H' \mid \epsilon \mid A,\Sigma,\mathrm{nnf}(\neg T)$. Thus $[\Box A,\Diamond\Sigma] \notin H$.

5.3.16 THEOREM $(\mathsf{K}+T)^{\mathcal{S},3}$: termination

Backward proof search in $(\mathsf{K}+T)^{\mathcal{S},3}$ for a sequent of the form $\epsilon \mid \epsilon \mid A,\mathrm{nnf}(\neg T)$ always terminates.

Proof

Assume that we do backward proof search for $\epsilon \mid \epsilon \mid A,\mathrm{nnf}(\neg T)$ in $(\mathsf{K}+T)^{\mathcal{S},3}$. When $(\lor)$, $(\land)$, $(\Diamond,\mathrm{new})$, $(\Diamond,\mathrm{dup})$ are applied backwards, then at least one connective is removed from the right

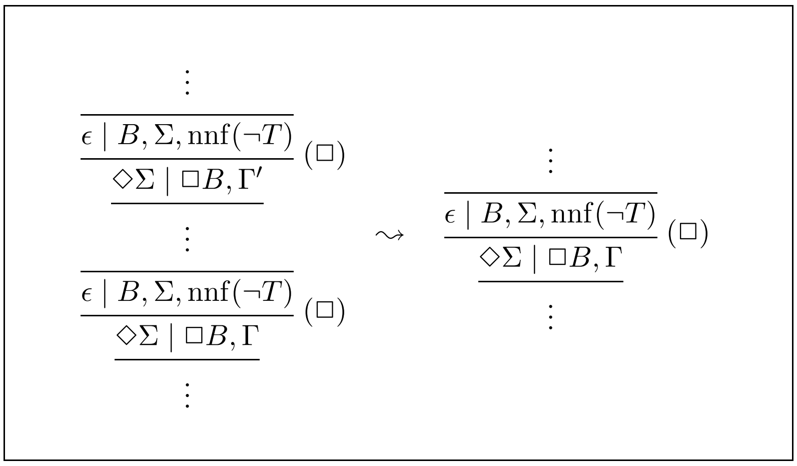

$\epsilon \mid B, \Sigma, \mathrm{nnf}(\neg T)$
$\Diamond\Sigma \mid \Box B, \Gamma'$
$(\Box)$
$\epsilon \mid B, \Sigma, \mathrm{nnf}(\neg T)$
$\Diamond\Sigma \mid \Box B, \Gamma$
$(\Box)$
$\rightsquigarrow$
$\epsilon \mid B, \Sigma, \mathrm{nnf}(\neg T)$
$\Diamond\Sigma \mid \Box B, \Gamma$
$(\Box)$

Figure 5.e: Removing loops from proofs in $\mathsf{K}+T^{\mathcal{S}}$. This transformation is used in the proof of theorem 5.3.15.

hand side of the sequent. Therefore there must be at least one backward application of $(\Box)$ in a loop. However, because of the loop check, no two premises of a $(\Box)$ application on one branch of a proof can be equal. Since Σ does not contain duplicate elements, there are only finitely many possibilities for such premises.

This leads to the following measure: $m(H \mid \Diamond\Sigma \mid \Gamma) := (c \cdot 2^c - \mathrm{card}(H)) \cdot c + \mathrm{length}(\Diamond\Sigma) + \mathrm{length}(\Gamma) - \mathrm{card}(\Diamond\Sigma)$, where $c := \mathrm{length}(A) + \mathrm{length}(\mathrm{nnf}(\neg T))$.

All elements of H are multisets of the form $\Box B, \Diamond\Sigma$. Because of the subformula property of $(\mathsf{K}+T)^{\mathcal{S},3}$ and since $\Diamond\Sigma$ contains no duplicate elements, we know that H contains at most $c \cdot 2^c$ elements. Therefore always $m(H \mid \Gamma) \geq 0$ during backward proof search.

The value computed by m decreases with each backward application of a rule of $\mathsf{K}^{\mathcal{S},3}$:

1. $m(H \mid \Diamond\Sigma \mid A \vee B, \Gamma) > m(H \mid \Diamond\Sigma \mid A, B, \Gamma)$
2. $m(H \mid \Diamond\Sigma \mid A \wedge B, \Gamma) > m(H \mid \Diamond\Sigma \mid A, \Gamma)$, $m(H \mid \Diamond\Sigma \mid A \wedge B, \Gamma) > m(H \mid \Diamond\Sigma \mid B, \Gamma)$
3. $m(H \mid \Diamond A, \Diamond\Sigma \mid \Gamma) > m(H \mid \Diamond\Sigma \mid \Diamond A, \Gamma)$
4. $m(H \mid \Diamond\Sigma \mid \Gamma) > m(H \mid \Diamond\Sigma \mid \Diamond A, \Gamma)$
5. The only interesting case is the $(\Box)$ rule. Assume that $[\Box A, \Diamond\Sigma] \notin H$ and that $\Diamond\Sigma$ contains no duplicate elements. Then we have:

$$\begin{aligned}
&m(H \mid \Diamond\Sigma \mid \Box A, \Gamma) - m([\Box A, \Diamond\Sigma], H \mid \epsilon \mid A, \Sigma, \mathrm{nnf}(\neg T)) \\
&\quad = ((c \cdot 2^c - \mathrm{card}(H)) - (c \cdot 2^c - \mathrm{card}([\Box A, \Diamond\Sigma], H))) \cdot c \\
&\qquad + \mathrm{length}(\Box A, \Gamma) + \mathrm{length}(\Diamond\Sigma) - \mathrm{card}(\Diamond\Sigma) - \mathrm{length}(A, \Sigma, \mathrm{nnf}(\neg T)) \\
&\quad = 1 \cdot c + 1 + \mathrm{length}(\Gamma) - \mathrm{length}(\mathrm{nnf}(\neg T)) \\
&\quad \geq 1 + \mathrm{length}(\Gamma) \\
&\quad > 0.
\end{aligned}$$

5.3.17 THEOREM proofs in $(\mathsf{K}+T)^{\mathcal{S},3}$

If $(\mathsf{K}+T)^{\mathcal{S},3} \vdash \epsilon \mid \epsilon \mid A, \mathrm{nnf}(\neg T)$, then there exists a proof of $\epsilon \mid \epsilon \mid A, \mathrm{nnf}(\neg T)$ in $(\mathsf{K}+T)^{\mathcal{S},3}$ such that $(\Box)$ is only applied backwards if no other rule is applicable backwards.

Proof

Because of theorem 5.2.12 and theorem 5.3.15 there exists a proof of $\epsilon \mid A, \mathrm{nnf}(\neg T)$ in $(\mathsf{K}+T)^{\mathcal{S}}$ such that $(\Box)$ is only applied backwards if no other rule is applicable backwards. We now use the construction from the proof of theorem 5.3.15 to transform it into a proof of $\epsilon \mid \epsilon \mid A, \mathrm{nnf}(\neg T)$ in $(\mathsf{K}+T)^{\mathcal{S},3}$. It can be easily checked that this proof has the desired property.

5.3.18 THEOREM $(\mathsf{K}+T)^{\mathcal{S},3}$: permutation of rules

For the calculus $(\mathsf{K}+T)^{\mathcal{S},3}$, the following applies for all axioms x and all rules r:

- If r is not $(\Box)$, then $x \rightsquigarrow rx$.
- If neither r nor s is $(\Box)$, then $rs\cdots \rightsquigarrow sr\cdots$.

Proof

Analogous to the corresponding proof for $(\mathsf{K}+T)^{\mathcal{S},2}$, as the history has only an effect in the $(\Box)$ rule.

5.3.19 REMARK invertible rules in $(\mathsf{K}+T)^{\mathcal{S},3}$

Note that the rule $(\Diamond, \mathrm{new})$ of $(\mathsf{K}+T)^{\mathcal{S},3}$ is not invertible. We have $(\mathsf{K}+T)^{\mathcal{S},3} \vdash [\Box\mathsf{true}, \Diamond p_1] \mid \epsilon \mid \Box\mathsf{true}, \Diamond p_1$, as the following proof shows:

$$\dfrac{\dfrac{}{[\Box\mathsf{true}], [\Box\mathsf{true}, \Diamond p_1] \mid \epsilon \mid \mathsf{true}}\,(\mathsf{true})}{[\Box\mathsf{true}, \Diamond p_1] \mid \epsilon \mid \Box\mathsf{true}, \Diamond p_1}\,(\Box)$$

However, $(\mathsf{K}+T)^{\mathcal{S},3} \nvdash [\Box\mathsf{true}, \Diamond p_1] \mid \Diamond p_1 \mid \Box\mathsf{true}$, since the $(\Box)$ rule is not applicable backwards on this sequent.

The two theorems 5.3.17 and 5.3.18 show that if we do backward proof search in $(\mathsf{K}+T)^{\mathcal{S},3}$ for a sequent of the form $\epsilon \mid \epsilon \mid A, \mathrm{nnf}(\neg T)$, then we can do this as if the rules $(\vee)$, $(\wedge)$, $(\Diamond, \mathrm{new})$, $(\Diamond, \mathrm{dup})$ were invertible: We apply these rules backwards as long as possible, and the order in which we do this does not matter.

5.3.20 THEOREM $(\mathsf{K}+T)^{\mathcal{S},3}$: empty the history

$$(\mathsf{K}+T)^{\mathcal{S},3} \vdash H \mid \Diamond\Sigma \mid \Gamma \quad \Rightarrow \quad (\mathsf{K}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma$$

Proof

Let $\mathcal{P}$ be the proof $H \mid \Diamond\Sigma \mid \Gamma$ in $(\mathsf{K}+T)^{\mathcal{S},3}$. Let $\Box A'$ be a formula and $\Diamond\Sigma'$ a multiset of formulas. With an induction on proof depth we can show that then $(\mathsf{K}+T)^{\mathcal{S},3} \vdash f(H) \mid \Diamond\Sigma \mid \Gamma$, where $f(H) = H$ if $[\Box A', \Diamond\Sigma'] \notin H$ and $H = f(H), [\Box A', \Diamond\Sigma']$ if $[\Box A', \Diamond\Sigma'] \in H$. With an induction on the number of elements in H we obtain $(\mathsf{K}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma$.

5.3.21 THEOREM $(\mathsf{K}+T)^{\mathcal{S},3}$: extend (id)

If A is in negation normal form, then:

$$(\mathsf{K}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid A, \mathrm{nnf}(\neg A), \Gamma$$

Proof

With an induction on length(A) we prove $(\mathsf{K}+T)^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, \mathrm{nnf}(\neg A), \Gamma$. The only difference between this proof and the corresponding proof for $\mathsf{K}^{\mathcal{S},2}$ (theorem 5.2.16) is the multiset $\mathrm{nnf}(\neg T)$ in the premise of $(\Box)$ applications. Therefore we only show the case $A \equiv \Box B$.

1. $A \equiv \Box B$: If $B \notin \Sigma$, then we obtain $\mathsf{K}^{\mathcal{S},2} \vdash \epsilon \mid B, \mathrm{nnf}(\neg B), \Sigma, \mathrm{nnf}(\neg T)$ with the induction hypothesis, and with applications of $(\Box)$ and $(\Diamond, \mathrm{new})$ follows $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Box B, \Diamond \mathrm{nnf}(\neg B)$. If $B \in \Sigma$, then we obtain $\mathsf{K}^{\mathcal{S},2} \vdash \epsilon \mid B, \Sigma$ with the induction hypothesis, and with applications of $(\Box)$ and $(\Diamond, \mathrm{dup})$ follows $\mathsf{K}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Box B, \Diamond \mathrm{nnf}(\neg B)$.

Now we use theorem 5.3.15 to obtain $(\mathsf{K}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid A, \mathrm{nnf}(\neg A), \Gamma$.

5.3.22 THEOREM $(\mathsf{K}+T)^{\mathcal{S},3}$: duplicate formulas

$$(\mathsf{K}+T)^{\mathcal{S},3} \vdash H \mid \Diamond\Sigma \mid A, A, \Gamma \quad \Rightarrow \quad (\mathsf{K}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid A, \Gamma$$

Proof

The idea is to go to the calculus $(\mathsf{K}+T)^{\mathcal{S},2}$, to prove the corresponding theorem for this calculus and then return to $(\mathsf{K}+T)^{\mathcal{S},3}$ with the help of theorem 5.3.15.

With theorem 5.3.20 we obtain $(\mathsf{K}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid A, A, \Gamma$ and with theorem 5.3.15 follows $(\mathsf{K}+T)^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, A, \Gamma$. Thus $(\mathsf{K}+T)^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, \Gamma$ because of theorem 5.3.10. With theorem 5.3.15 we finally obtain $(\mathsf{K}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid A, \Gamma$.

5.3.23 THEOREM $(\mathsf{K}+T)^{\mathcal{S},3}$: use-check

Let $\mathcal{P}$ be a proof in $(\mathsf{K}+T)^{\mathcal{S},3}$ of the sequent $H \mid A, \Gamma$. We mark it as follows:

- The main formulas in the instances of axioms are marked.
- The main formula in the instances of the $(\Box)$ rule is marked.
- If a side formula is marked, then the main formula is marked.
- If in the premise of an instance of $(\vee)$, $(\wedge)$, $(\Diamond, \mathrm{new})$ or $(\Diamond, \mathrm{dup})$ a formula in Γ is marked, then the corresponding formula in the multiset Γ in the conclusion is marked.
- If in the premise of an instance of $(\vee)$, $(\wedge)$, $(\Diamond, \mathrm{new})$ or $(\Diamond, \mathrm{dup})$ a formula in $\Diamond\Sigma$ is marked, then the corresponding formula in the multiset $\Diamond\Sigma$ in the conclusion is marked.
- If in the premise of an instance of $(\Box)$ a formula in Σ is marked, then the corresponding formula in the multiset $\Diamond\Sigma$ in the conclusion is marked.

If the formula A in the sequent $H \mid A, \Gamma$ is not marked, then $(\mathsf{K}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Gamma$.

Proof

The idea is the same as in the proof of the preceding theorem: We prove the corresponding theorem for $(\mathsf{K}+T)^{\mathcal{S},2}$ and use theorem 5.3.15.

First we prove the theorem for the calculus $(\mathsf{K}+T)^{\mathcal{S},2}$, defining marked proofs in $(\mathsf{K}+T)^{\mathcal{S},2}$ exactly as described above for $(\mathsf{K}+T)^{\mathcal{S},3}$. Except for the multiset $\mathrm{nnf}(\neg T)$ in the premises of $(\Box)$ applications this is the same proof as the one of theorem 5.2.17.

Now we take the marked proof $\mathcal{P}$ of $H \mid \Diamond\Sigma \mid A, \Gamma$ and omit all histories, but do not change the marks (cp. the '$\Leftarrow$' part of the proof of theorem 5.3.15). We call the result $\mathcal{P}'$. With an induction on proof depth we can easily show that $\mathcal{P}'$ is a marked proof of $\Diamond\Sigma \mid A, \Gamma$ in $(\mathsf{K}+T)^{\mathcal{S},2}$. Thus A is not marked, and therefore $(\mathsf{K}+T)^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$. With theorem 5.3.15 we obtain $(\mathsf{K}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma$.

See also the remark 6.3.3 for the effects of use-check on the efficiency of backward proof search.

5.3.24 THEOREM (empty) is admissible

If $\epsilon \mid \Diamond\Sigma \mid \Gamma$ is provable in the calculus $(\mathsf{K}+T)^{\mathcal{S},3}$ plus the rule $\frac{\epsilon|\Diamond\Sigma|\Gamma}{H|\Diamond\Sigma|\Gamma}$(empty), then $(\mathsf{K}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma$.

Proof

Let $\mathcal{P}$ be a proof of $\epsilon \mid \Diamond\Sigma \mid \Gamma$ in $(\mathsf{K}+T)^{\mathcal{S},3}$ plus the rule (empty). If we remove the applications of (empty) and all histories, then we obtain a proof of $\Diamond\Sigma \mid \Gamma$ in $(\mathsf{K}+T)^{\mathcal{S},2}$, i.e. $(\mathsf{K}+T)^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$ (induction on proof depth). With theorem 5.3.15 follows $(\mathsf{K}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma$.

5.3.25 REMARK using optimisations during proof search in $(\mathsf{K}+T)^{\mathcal{S},3}$

Note that in theorem 5.3.21 we write $(\mathsf{K}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid A, \mathrm{nnf}(\neg A), \Gamma$ and not $(\mathsf{K}+T)^{\mathcal{S},3} \vdash H \mid \Diamond\Sigma \mid A, \mathrm{nnf}(\neg A), \Gamma$, since for example $(\mathsf{K}+T)^{\mathcal{S},3} \nvdash [\Box p_1, \Diamond\neg p_1] \mid \epsilon \mid \Box p_1, \Diamond\neg p_1$. Also in the theorems 5.3.22 and 5.3.23 we have empty histories. During backward proof search, however, the history will in general not be empty, i.e. we cannot apply these theorems directly.

This problem is solved by theorem 5.3.24. If we obtain a sequent $H \mid \Diamond\Sigma \mid \Gamma$ during backward proof search and we know that the sequent $\epsilon \mid \Diamond\Sigma \mid \Gamma$ is provable in $(\mathsf{K}+T)^{\mathcal{S},3}$, then we simply insert a backward application of the (empty) rule. If the whole search is successful, then we obtain a proof in $(\mathsf{K}+T)^{\mathcal{S},3}$ plus the rule (empty). Provided that we did backward proof search for a sequent of the form $\epsilon \mid \epsilon \mid B$, we can now apply theorem 5.3.24.

Example

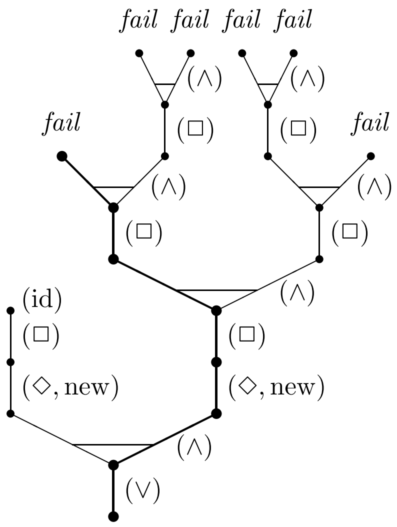
We do backward proof search in $(\mathsf{K}+T)^{\mathcal{S},3}$ for the sequent $\epsilon \mid \epsilon \mid \Box(\Box p_0 \vee \Diamond\neg p_0) \vee \Diamond p_2$. After four backward applications of rules, we have the following situation:

$$\dfrac{\dfrac{\dfrac{\dfrac{[\Box(\Box p_0 \vee \Diamond\neg p_0), \Diamond p_2] \mid \epsilon \mid \Box p_0, \Diamond\neg p_0, p_2}{[\Box(\Box p_0 \vee \Diamond\neg p_0), \Diamond p_2] \mid \epsilon \mid \Box p_0 \vee \Diamond\neg p_0, p_2}(\vee)}{\epsilon \mid \Diamond p_2 \mid \Box(\Box p_0 \vee \Diamond\neg p_0)}(\Box)}{\epsilon \mid \epsilon \mid \Box(\Box p_0 \vee \Diamond\neg p_0), \Diamond p_2}(\Diamond, \mathrm{new})}{\epsilon \mid \epsilon \mid \Box(\Box p_0 \vee \Diamond\neg p_0) \vee \Diamond p_2}(\vee)$$

Since the right hand side of the sequent $[\Box(\Box p_0 \vee \Diamond\neg p_0), \Diamond p_2] \mid \epsilon \mid \Box p_0, \Diamond\neg p_0, p_2$ is of the form $A, \mathrm{nnf}(\neg A), \Gamma$, we apply (empty) backwards to obtain the sequent $\epsilon \mid \epsilon \mid \Box p_0, \Diamond\neg p_0, p_2$ and then know because of theorem 5.3.21 that this sequent is provable in $(\mathsf{K}+T)^{\mathcal{S},3}$. We end up with the following proof of $\epsilon \mid \epsilon \mid \Box(\Box p_0 \vee \Diamond\neg p_0) \vee \Diamond p_2$ in $(\mathsf{K}+T)^{\mathcal{S},3}$ plus the rule (empty):

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathcal{P}}{\epsilon \mid \epsilon \mid \Box p_0, \Diamond\neg p_0, p_2}}{[\Box(\Box p_0 \vee \Diamond\neg p_0), \Diamond p_2] \mid \epsilon \mid \Box p_0, \Diamond\neg p_0, p_2}(\mathrm{empty})}{[\Box(\Box p_0 \vee \Diamond\neg p_0), \Diamond p_2] \mid \epsilon \mid \Box p_0 \vee \Diamond\neg p_0, p_2}(\vee)}{\epsilon \mid \Diamond p_2 \mid \Box(\Box p_0 \vee \Diamond\neg p_0)}(\Box)}{\epsilon \mid \epsilon \mid \Box(\Box p_0 \vee \Diamond\neg p_0), \Diamond p_2}(\Diamond, \mathrm{new})}{\epsilon \mid \epsilon \mid \Box(\Box p_0 \vee \Diamond\neg p_0) \vee \Diamond p_2}(\vee)$$

Note that we do not construct the proof $\mathcal{P}$ of the sequent $\epsilon \mid \epsilon \mid \Box p_0, \Diamond\neg p_0$ in $(\mathsf{K}+T)^{\mathcal{S},3}$, but because of theorem 5.3.21 we know that such a proof exists. With theorem 5.3.24 follows $(\mathsf{K}+T)^{\mathcal{S},3} \vdash \epsilon \mid \epsilon \mid \Box(\Box p_0 \vee \Diamond\neg p_0) \vee \Diamond p_2$.



At first we apply $(\vee)$ backwards. One branch caused by the following backward application of $(\wedge)$ ends in an instance of the (id) axiom. The other branch splits up in six subbranches, which all fail. Note that in this example there is no possibility for backtracking. Therefore $(\mathsf{K}+T)^{\mathcal{S},3} \vdash \Diamond p_0 \vee p_1, \mathrm{nnf}(\neg T)$. The thick lines correspond to the leftmost model in figure 5.h.

Figure 5.f: The search tree of the example in remark 5.3.26. See figure 5.g for a part of the corresponding non-proof, and figure 5.h for the six countermodels that correspond to the six failing branches.

5.3.26 REMARK $(\mathsf{K}+T)^{\mathcal{S},3}$: backward proof search

Assume that the backward proof search fails. As for K we can extract a countermodel from the search tree. First we choose a subtree exactly as we did for example for $\mathsf{K}^{\mathcal{S},2}$. In contrast to $\mathsf{K}^{\mathcal{S},2}$, we now have nodes marked with *fail* that failed because of the loop check in the $(\Box)$ rule of $(\mathsf{K}+T)^{\mathcal{S},3}$. If we cannot apply $(\Box)$ backwards because $[\Box A, \Diamond\Sigma] \in H$, then the sequent $A, \Sigma, \mathrm{nnf}(\neg T)$ must have occurred before on this branch. Therefore we obtain a loop in the countermodel for each such node.

Example

Backward proof search for $\Diamond p_0 \vee p_1$ in $(\mathsf{K}+\{\Diamond p_0 \vee \Diamond p_1\})^{\mathcal{S},3}$ terminates, in contrast to proof search in $(\mathsf{K}+\{\Diamond p_0 \vee \Diamond p_1\})^{\mathcal{S},2}$ (cp. the example in remark 5.3.12). Note that in this example we have no possibility for backtracking, since we always have at most one possible way to apply $(\Box)$ backwards. Therefore the extracted countermodels consist of just one branch.

See figure 5.f for the search tree and 5.g for a part of the corresponding non-proof. Figure 5.h shows the six countermodels that correspond to the six failed branches of the search tree.

5.3.27 REMARK size of elements of the history

The history in the sequents in $(\mathsf{K}+T)^{\mathcal{S},3}$ is a multiset of multisets of formulas, whereas for $\mathsf{S4}$ there is a calculus where the history is just a multiset of formulas. Why this difference? From theorem 6.3.1 follows that in $\mathsf{K}+T$ branches of countermodels must sometimes have exponential length (with respect to $\mathrm{length}(A) + \mathrm{length}(T)$). If we do backward proof search in a sequent calculus for $\mathsf{K}+T$ and it is possible to extract countermodels from a failed search 'in the usual way', then there are branches in the search tree with exponential length, i.e. a history can contain exponentially many different elements. Therefore the history cannot be a multiset of subformulas of A and T, since there are at most $\mathrm{length}(A) + \mathrm{length}(T)$ different subformulas of A and T.

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{}{[\Box\neg p_0, \Diamond p_0] \mid \epsilon \mid \neg p_0, p_0, A}\,(\mathrm{id})
      }{\epsilon \mid \Diamond p_0 \mid p_1, \Box\neg p_0}\,(\Box)
    }{\epsilon \mid \epsilon \mid \Diamond p_0, p_1, \Box\neg p_0}\,(\Diamond, \mathrm{new})
    \quad
    \dfrac{
      \dfrac{
        \dfrac{
          \dfrac{
            \dfrac{
              [\Box\neg p_0], \Pi \mid \epsilon \mid \neg p_0, \Box\neg p_0
              \quad
              \dfrac{
                \dfrac{
                  [\Box\neg p_1], [\Box\neg p_0], \Pi \mid \epsilon \mid \neg p_1, \Box\neg p_0
                  \quad
                  [\Box\neg p_1], [\Box\neg p_0], \Pi \mid \epsilon \mid \neg p_1, \Box\neg p_1
                }{[\Box\neg p_1], [\Box\neg p_0], \Pi \mid \epsilon \mid \neg p_1, A}\,(\wedge)
              }{[\Box\neg p_0], \Pi \mid \epsilon \mid \neg p_0, \Box\neg p_1}\,(\Box)
            }{[\Box\neg p_0], \Pi \mid \epsilon \mid \neg p_0, A}\,(\wedge)
          }{\Pi \mid \epsilon \mid \neg p_1, p_0, \Box\neg p_0}\,(\Box)
          \quad
          \vdots
        }{\Pi \mid \epsilon \mid \neg p_1, p_0, A}\,(\wedge)
      }{\epsilon \mid \Diamond p_0 \mid p_1, \Box\neg p_1}\,(\Box)
    }{\epsilon \mid \epsilon \mid \Diamond p_0, p_1, \Box\neg p_1}\,(\Diamond, \mathrm{new})
  }{\epsilon \mid \epsilon \mid \Diamond p_0, p_1, A}\,(\wedge)
}{\epsilon \mid \epsilon \mid \Diamond p_0 \vee p_1, A}\,(\vee)
$$

Figure 5.g: A part of the nonproof of the example in 5.3.26. We use the abbreviations $A \equiv \Box\neg p_0 \wedge \Box\neg p_1$ and $\Pi = [\Box\neg p_1, \Diamond p_0]$. See figure 5.f for the corresponding search tree.

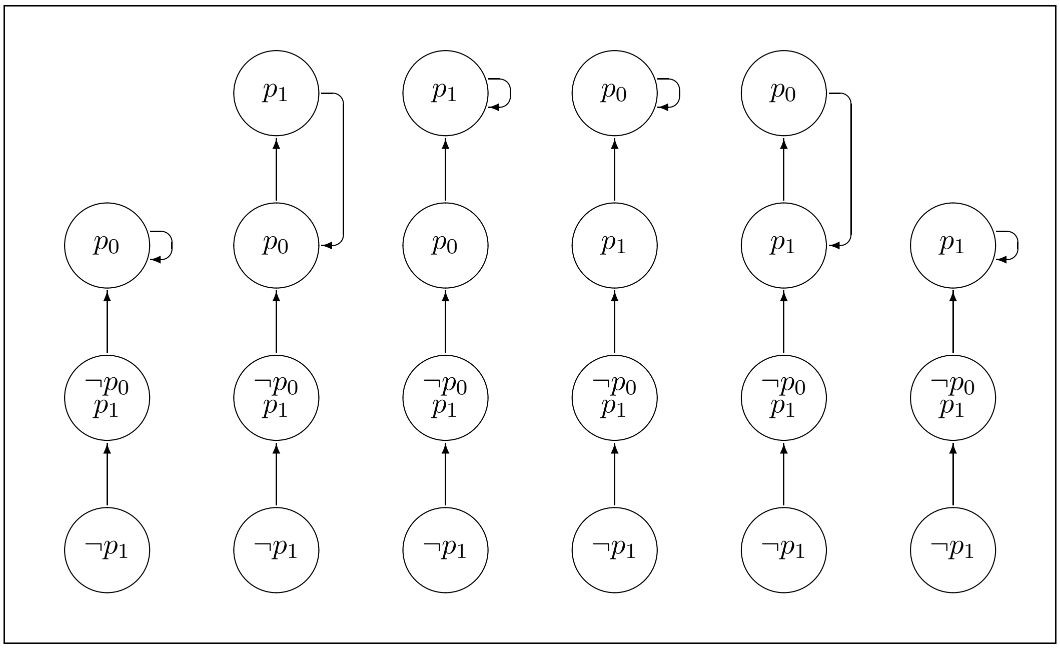


Figure 5.h: The six countermodels of the example in 5.3.26. They correspond to the six failing branches of the search tree in figure 5.f.

5.3.28 REMARK $\mathsf{K}+T$ and PLTL

The models in figure 5.h remind one of PLTL models. The calculus $(\mathsf{K}+T)^{\mathcal{S},3}$ is designed for a depth-first backward proof search. In the $(\Box)$ rule we check only whether the current sequent already occurred before on the same branch. However, the same states can occur frequently on different branches. Therefore it is possible that a search method as for PLTL as the one used in [Gou89], [Jan90] and [Wol85], where the current state is compared with all other states computed so far, is advantageous.

5.4 KT

5.4.1 REMARK from $\mathsf{KT}^{\mathcal{G},2}$ to $\mathsf{KT}^{\mathcal{S}}$

This is the same step as the one from $\mathsf{K}^{\mathcal{G},2}$ to $\mathsf{K}^{\mathcal{S}}$ (cp. remark 5.2.1). Since no rule of $\mathsf{K}^{\mathcal{G},2}$ influences a node that is nearer to the root than the current node, we can forget everything about the worlds ‘below’ the current world.

In contrast to the calculus for K, we have the formula A in the premise of the $(\Diamond)$ rule.

5.4.2 DEFINITION sequent calculus $\mathsf{KT}^{\mathcal{S}}$

axioms:

$$\frac{}{\Diamond\Sigma \mid \mathsf{true}, \Gamma}\ (\mathsf{true}) \qquad\qquad \frac{}{\Diamond\Sigma \mid P, \neg P, \Gamma}\ (\mathrm{id})$$

rules:

$$\frac{\Diamond\Sigma \mid A, B, \Gamma}{\Diamond\Sigma \mid A \vee B, \Gamma}\ (\vee) \qquad\qquad \frac{\Diamond\Sigma \mid A, \Gamma \quad \Diamond\Sigma \mid B, \Gamma}{\Diamond\Sigma \mid A \wedge B, \Gamma}\ (\wedge)$$

$$\frac{\Diamond A, \Diamond\Sigma \mid A, \Gamma}{\Diamond\Sigma \mid \Diamond A, \Gamma}\ (\Diamond) \qquad\qquad \frac{\epsilon \mid A, \Sigma}{\Diamond\Sigma \mid \Box A, \Gamma}\ (\Box)$$

main formulas: true in (true), P and $\neg P$ in (id), $A \vee B$ in ($\vee$), $A \wedge B$ in ($\wedge$), $\Diamond A$ in ($\Diamond$), $\Box A$ in ($\Box$)

Example

$\mathsf{KT}^{\mathcal{S}} \vdash \epsilon \mid \Diamond\Diamond\Box(p_0 \vee \Box\Diamond\neg p_0)$ as the following proof shows. We use the abbreviation $A \equiv \Box(p_0 \vee \Box\Diamond\neg p_0)$. See also theorem 6.4.4.

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{}{\epsilon \mid p_0, \Box\Diamond\neg p_0, \neg p_0}\ (\text{id})}{\epsilon \mid p_0 \vee \Box\Diamond\neg p_0, \neg p_0}\ (\vee)}{\Diamond\neg p_0 \mid \neg p_0, A}\ (\Box)}{\epsilon \mid \Diamond\neg p_0, A}\ (\Diamond)}{\Diamond A \mid p_0, \Box\Diamond\neg p_0, A, A}\ (\Box)}{\epsilon \mid p_0, \Box\Diamond\neg p_0, A, \Diamond A}\ (\Diamond)}{\epsilon \mid p_0 \vee \Box\Diamond\neg p_0, A, \Diamond A}\ (\vee)}{\Diamond A, \Diamond\Diamond A \mid A}\ (\Box)}{\Diamond\Diamond A \mid \Diamond A}\ (\Diamond)}{\epsilon \mid \Diamond\Diamond A}\ (\Diamond)$$

Note that $\mathsf{KT}^{\mathcal{S}} \nvdash \epsilon \mid \Diamond\Box(p_0 \vee \Box\Diamond\neg p_0)$, i.e. $\mathsf{KT} \models \Diamond\Diamond A$, but $\mathsf{KT} \not\models \Diamond A$.

5.4.3 THEOREM $\mathsf{KT}^{\mathcal{S}}$: weakening

If $\mathsf{KT}^{\mathcal{S}} \vdash \Diamond\Sigma \mid \Gamma$, then $\mathsf{KT}^{\mathcal{S}} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$. Moreover, if d is the depth of the proof of $\Diamond\Sigma \mid \Gamma$, then there exists a proof of $\Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$ whose depth is at most d.

Proof

We make an induction on proof depth. The cases where the last step is an instance of an axiom or an application of ($\vee$), ($\wedge$) or ($\Box$) are the same as in the corresponding proof for $\mathsf{K}^{\mathcal{S}}$ (theorem 5.2.3). We only show the case with ($\Diamond$).

1. The last step is $\frac{\Diamond A, \Diamond\Sigma \mid A, \Gamma}{\Diamond\Sigma \mid \Diamond A, \Gamma}(\Diamond)$: We obtain $\mathsf{KT}^{\mathcal{S}} \vdash \Diamond\Delta, \Diamond A, \Diamond\Sigma \mid \Pi, A, \Gamma$ with the induction hypothesis, and with an application of ($\Diamond$) follows $\mathsf{KT}^{\mathcal{S}} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, \Diamond A, \Gamma$.

5.4.4 THEOREM $\mathsf{KT}^{\mathcal{S}}$: invertible rules

The rule ($\Box$) of $\mathsf{KT}^{\mathcal{S}}$ is not invertible. All the other rules are strongly invertible.

Proof

Analogous to the corresponding proof for $\mathsf{K}^{\mathcal{S}}$ (theorem 5.2.4). The only interesting case is $(\Box) \cdots \rightsquigarrow (\Diamond)(\Box)$. To prove it, we use the transformation below plus weakening.

$$\frac{\epsilon \mid A, \Sigma}{\Diamond\Sigma \mid \Box A, \Diamond B, \Gamma}\ (\Box) \quad\rightsquigarrow\quad \cfrac{\cfrac{\epsilon \mid A, B, \Sigma}{\Diamond B, \Diamond\Sigma \mid \Box A, B, \Gamma}\ (\Box)}{\Diamond\Sigma \mid \Box A, \Diamond B, \Gamma}\ (\Diamond)$$

5.4.5 THEOREM $\mathsf{KT}^{\mathcal{S}}$: duplicate formulas

$$\begin{aligned}\mathsf{KT}^{\mathcal{S}} \vdash \Diamond\Sigma \mid A, A, \Gamma \quad &\Rightarrow \quad \mathsf{KT}^{\mathcal{S}} \vdash \Diamond\Sigma \mid A, \Gamma \\ \mathsf{KT}^{\mathcal{S}} \vdash \Diamond A, \Diamond A, \Diamond\Sigma \mid \Gamma \quad &\Rightarrow \quad \mathsf{KT}^{\mathcal{S}} \vdash \Diamond A, \Diamond\Sigma \mid \Gamma\end{aligned}$$

Proof

The proof is analogous to the corresponding proof for $\mathsf{K}^{\mathcal{S}}$ (theorem 5.2.6). We show only the case where the last step is $(\Diamond)$ application with the duplicate formula as the main formula.

1. The last step is $\frac{\Diamond B, \Diamond\Sigma \mid B, \Diamond B, \Gamma}{\Diamond\Sigma \mid \Diamond B, \Diamond B, \Gamma}(\Diamond)$, where $A \equiv \Diamond B$: Let d be the depth of the proof of $\Diamond B, \Diamond\Sigma \mid B, \Diamond B, \Gamma$. Since $(\Diamond)$ is strongly invertible we know that $\mathsf{K}^{\mathcal{S}} \vdash \Diamond B, \Diamond B, \Diamond\Sigma \mid B, B, \Gamma$ with a proof of depth $\leq d-1$. Using the induction hypothesis twice we obtain $\mathsf{KT}^{\mathcal{S}} \vdash \Diamond B, \Diamond\Sigma \mid B, \Gamma$ with a proof of depth $\leq d-1$, and with an application of $(\Diamond)$ follows $\mathsf{KT}^{\mathcal{S}} \vdash \Diamond\Sigma \mid \Diamond B, \Gamma$ with a proof of depth $\leq d$.

5.4.6 THEOREM equivalence of $\mathsf{KT}^{\mathcal{G},2}$ and $\mathsf{KT}^{\mathcal{S}}$

$$\mathsf{KT}^{\mathcal{G},2} \vdash \boxed{\boxed{\epsilon \mid A}} \quad \Leftrightarrow \quad \mathsf{KT}^{\mathcal{S}} \vdash \epsilon \mid A$$

Proof

'$\Rightarrow$':

We can proceed exactly as in the corresponding proof of the equivalence of $\mathsf{K}^{\mathcal{G},2}$ and $\mathsf{K}^{\mathcal{S}}$ (theorem 5.2.7). We can easily check that the step by step translation at the end of the proof is still possible.

'$\Leftarrow$':

As for K (cp. theorem 5.2.7) we can translate the proof step by step (beginning at the root).

5.4.7 THEOREM proofs in $\mathsf{KT}^{\mathcal{H}}$ from proofs in $\mathsf{KT}^{\mathcal{S}}$

$$\mathsf{KT}^{\mathcal{S}} \vdash \epsilon \mid \mathrm{nnf}(A) \quad \Rightarrow \quad \mathsf{KT}^{\mathcal{H}} \vdash A$$

Proof

We can reuse most parts of the proof of 5.2.8.

In addition we have to prove that $\mathsf{KT}^{\mathcal{H}} \vdash \Diamond C \vee \Diamond A_1 \vee \ldots \vee \Diamond A_m \vee C \vee B_2 \vee \ldots \vee B_n$ implies $\mathsf{KT}^{\mathcal{H}} \vdash \Diamond A_1 \vee \ldots \vee \Diamond A_m \vee \Diamond C \vee B_2 \vee \ldots \vee B_n$. This follows from the proof fragment in figure 5.i.

5.4.8 REMARK two multisets per sequent

Usually a sequent in a sequent calculus consists of only one multiset. One way to look at the two multisets in the sequent in $\mathsf{KT}^{\mathcal{S}}$ is to consider the formulas on the left hand side as marked formulas. This mark prevents repeated backward application of the $(\Diamond)$ rule. In remark 5.4.9 we will see that this is important for termination.

$$\dfrac{}{\Box\neg C \to \neg C}\,(\mathrm{t})$$

$$\vdots \quad (\Diamond\Box_2), (\mathrm{cpc}), (\mathrm{mp})$$

$$\overline{C \to \Diamond C}$$

$$\vdots \quad (\mathrm{cpc}), (\mathrm{mp})$$

$$\overline{C \vee \Diamond C \to \Diamond C}$$

$$\vdots \quad (\mathrm{cpc}), (\mathrm{mp})$$

$$\dfrac{\dfrac{\vdots}{\begin{array}{l}\Diamond C \vee \Diamond A_1 \vee \ldots \vee \Diamond A_m \\ \vee\, C \vee B_2 \vee \ldots \vee B_n\end{array}} \qquad \dfrac{}{\begin{array}{l}(\Diamond C \vee \Diamond A_1 \vee \ldots \vee \Diamond A_m \vee C \vee B_2 \vee \ldots \vee B_n) \\ \to (\Diamond A_1 \vee \ldots \vee \Diamond A_m \vee \Diamond C \vee B_2 \vee \ldots \vee B_n)\end{array}}}{\Diamond A_1 \vee \ldots \vee \Diamond A_m \vee \Diamond C \vee B_2 \vee \ldots \vee B_n}\,(\mathrm{mp})$$

Figure 5.i: Simulating the $(\Diamond)$ rule of $\mathsf{KT}^{\mathcal{S}}$ in $\mathsf{KT}^{\mathcal{H}}$.

5.4.9 REMARK simplifying $\mathsf{KT}^{\mathcal{S}}$

If we simplify the calculus $\mathsf{KT}^{\mathcal{S}}$ in the same way as we simplified $\mathsf{K}^{\mathcal{S}}$ in remark 5.2.9, then we obtain the following calculus.

axioms:

$$\dfrac{}{\mathsf{true}, \Gamma}\,(\mathsf{true}) \qquad\qquad \dfrac{}{P, \neg P, \Gamma}\,(\mathrm{id})$$

rules:

$$\dfrac{A, B, \Gamma}{A \vee B, \Gamma}\,(\vee) \qquad\qquad \dfrac{A, \Gamma \quad B, \Gamma}{A \wedge B, \Gamma}\,(\wedge)$$

$$\dfrac{A, \Diamond A, \Gamma}{\Diamond A, \Gamma}\,(\Diamond) \qquad\qquad \dfrac{A, \Gamma}{\Box A, \Diamond\Gamma, \Delta}\,(\Box)$$

The only difference between this calculus and the corresponding sequent calculus for K is the additional $(\Diamond)$ rule. This rule contains a hidden contraction, and therefore backward proof search in the new calculus for KT does in general not terminate.

Example

Backward proof search for $\Diamond(p_0 \wedge p_2) \vee \neg p_0$ does not terminate in the calculus defined in this remark, although we never apply $(\Diamond)$ backwards on a formula $\Diamond A$ if A already occurs in the sequent.

$$\dfrac{\dfrac{\dfrac{}{p_0, \Diamond(p_0 \wedge p_2), \neg p_0}\,(\mathrm{id}) \quad \dfrac{\dfrac{\dfrac{\dfrac{}{p_2, p_0, \Diamond(p_0 \wedge p_2), \neg p_0}\,(\mathrm{id}) \quad \dfrac{\vdots}{p_2, p_2, \Diamond(p_0 \wedge p_2), \neg p_0}\,(\Diamond)}{p_2, p_0 \wedge p_2, \Diamond(p_0 \wedge p_2), \neg p_0}\,(\wedge)}{p_2, \Diamond(p_0 \wedge p_2), \neg p_0}\,(\Diamond)}{}}{p_0 \wedge p_2, \Diamond(p_0 \wedge p_2), \neg p_0}\,(\wedge)}{\dfrac{\Diamond(p_0 \wedge p_2), \neg p_0}{\Diamond(p_0 \wedge p_2) \vee \neg p_0}\,(\vee)}\,(\Diamond)$$

5.4.10 REMARK from $\mathsf{KT}^{\mathcal{S}}$ to $\mathsf{KT}^{\mathcal{S},2}$

The difference between $\mathsf{KT}^{\mathcal{S}}$ and $\mathsf{KT}^{\mathcal{S},2}$ is the same as the difference between $\mathsf{K}^{\mathcal{S}}$ and $\mathsf{K}^{\mathcal{S},2}$: We avoid duplicate $\Diamond$ formulas on the left hand side of the sequent. In contrast to K, this has important consequences for backward proof search (see theorem 6.4.4).

5.4.11 DEFINITION sequent calculus $\mathsf{KT}^{\mathcal{S},2}$

axioms:

$$\frac{}{\Diamond\Sigma \mid \mathsf{true}, \Gamma}\ (\mathsf{true}) \qquad\qquad \frac{}{\Diamond\Sigma \mid P, \neg P, \Gamma}\ (\mathrm{id})$$

rules:

$$\frac{\Diamond\Sigma \mid A, B, \Gamma}{\Diamond\Sigma \mid A \vee B, \Gamma}\ (\vee) \qquad\qquad \frac{\Diamond\Sigma \mid A, \Gamma \quad \Diamond\Sigma \mid B, \Gamma}{\Diamond\Sigma \mid A \wedge B, \Gamma}\ (\wedge)$$

$$\frac{\Diamond A, \Diamond\Sigma \mid A, \Gamma}{\Diamond\Sigma \mid \Diamond A, \Gamma}\ \Diamond A \notin \Diamond\Sigma \quad (\Diamond, \mathrm{new}) \qquad\qquad \frac{\Diamond\Sigma \mid A, \Gamma}{\Diamond\Sigma \mid \Diamond A, \Gamma}\ \Diamond A \in \Diamond\Sigma \quad (\Diamond, \mathrm{dup})$$

$$\frac{\epsilon \mid A, \Sigma}{\Diamond\Sigma \mid \Box A, \Gamma}\ (\Box)$$

main formulas: true in (true), P and $\neg P$ in (id), $A \vee B$ in $(\vee)$, $A \wedge B$ in $(\wedge)$, $\Diamond A$ in $(\Diamond, \mathrm{new})$ and $(\Diamond, \mathrm{dup})$, $\Box A$ in $(\Box)$
side formulas: A and B in $(\vee)$, A and B in $(\wedge)$, $\Diamond A$ and A in $(\Diamond, \mathrm{new})$, A in $(\Diamond, \mathrm{dup})$, A in $(\Box)$

5.4.12 THEOREM $\mathsf{KT}^{\mathcal{S},2}$: weakening

If $\mathsf{KT}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$, then $\mathsf{KT}^{\mathcal{S},2} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$. Moreover, if d is the depth of the proof of $\Diamond\Sigma \mid \Gamma$, then there exists a proof of $\Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$ whose depth is at most d.

Proof

Compared to the corresponding proof for $\mathsf{K}^{\mathcal{S},2}$ (theorem 5.2.11), only the cases for $(\Diamond, \mathrm{new})$ and $(\Diamond, \mathrm{dup})$ are different.

1. The last step is $\frac{\Diamond A, \Diamond\Sigma \mid A, \Gamma}{\Diamond\Sigma \mid \Diamond A, \Gamma}(\Diamond, \mathrm{new})$: If $\Diamond A \notin \Diamond\Delta$, then $\mathsf{KT}^{\mathcal{S},2} \vdash \Diamond\Delta, \Diamond A, \Diamond\Sigma \mid \Pi, A, \Gamma$ because of the induction hypothesis, and with a $(\Diamond, \mathrm{new})$ application follows $\mathsf{KT}^{\mathcal{S},2} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, \Diamond A, \Gamma$. If $\Diamond A \in \Diamond\Delta$, then $\mathsf{KT}^{\mathcal{S},2} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, A, \Gamma$ because of the induction hypothesis, and with a $(\Diamond, \mathrm{dup})$ application follows $\mathsf{KT}^{\mathcal{S},2} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, \Diamond A, \Gamma$.
2. The last step is $\frac{\Diamond\Sigma \mid \Gamma}{\Diamond\Sigma \mid \Diamond A, \Gamma}(\Diamond, \mathrm{dup})$: With the induction hypothesis we obtain $\mathsf{KT}^{\mathcal{S},2} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, \Diamond A, \Gamma$.

Note that if we removed the rule $(\Diamond, \mathrm{dup})$ from the calculus $\mathsf{KT}^{\mathcal{S},2}$, then this theorem would no longer be true, as e.g. $\epsilon \mid \Diamond(p_0 \vee \neg p_0)$ would be provable, but not $\Diamond(p_0 \vee \neg p_0) \mid \Diamond(p_0 \vee \neg p_0)$.

5.4.13 THEOREM $\mathsf{KT}^{\mathcal{S},2}$: invertible rules

The rule $(\Box)$ of $\mathsf{KT}^{\mathcal{S},2}$ is not invertible. All the other rules are strongly invertible.

Proof

Analogous to the proof of theorem 5.2.12. The interesting cases are $(\Box) \cdots \rightsquigarrow (\Diamond, \mathrm{new})(\Box) \cdots$ and $(\Box) \cdots \rightsquigarrow (\Diamond, \mathrm{dup})(\Box) \cdots$. To prove them, we use the first transformation below plus weakening and the second transformation, respectively.

$$\dfrac{\epsilon \mid A, \Sigma}{\Diamond\Sigma \mid \Box A, \Diamond B, \Gamma}(\Box) \quad \rightsquigarrow \quad \dfrac{\dfrac{\epsilon \mid A, B, \Sigma}{\Diamond B, \Diamond\Sigma \mid \Box A, B, \Gamma}(\Box)}{\Diamond\Sigma \mid \Box A, \Diamond B, \Gamma}(\Diamond, \text{new})$$

$$\dfrac{\dfrac{\vdots}{\epsilon \mid A, \Sigma}}{\Diamond\Sigma \mid \Box A, \Diamond B, \Gamma}(\Box) \quad \rightsquigarrow \quad \dfrac{\dfrac{\dfrac{\vdots}{\epsilon \mid A, \Sigma}}{\Diamond\Sigma \mid \Box A, B, \Gamma}(\Box)}{\Diamond\Sigma \mid \Box A, \Diamond B, \Gamma}(\Diamond, \text{dup})$$

5.4.14 THEOREM $\mathsf{KT}^{\mathcal{S},2}$: duplicate formulas

$$\mathsf{KT}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, A, \Gamma \quad \Rightarrow \quad \mathsf{KT}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, \Gamma$$

Proof

The proof is analogous to the corresponding proof for $\mathsf{KT}^{\mathcal{S}}$ (theorem 5.4.5). We show only those cases where the last step is a $(\Diamond, \text{new})$ or $(\Diamond, \text{dup})$ application with the duplicate formula as its main formula.

1. The last step is $\frac{\Diamond B, \Diamond\Sigma \mid B, \Diamond B, \Gamma}{\Diamond\Sigma \mid \Diamond B, \Diamond B, \Gamma}(\Diamond, \text{new})$, where $A \equiv \Diamond B$: Let d be the depth of the proof of $\Diamond B, \Diamond\Sigma \mid B, \Diamond B, \Gamma$. Since $(\Diamond, \text{dup})$ is strongly invertible we know that $\mathsf{KT}^{\mathcal{S},2} \vdash \Diamond B, \Diamond\Sigma \mid B, B, \Gamma$ with a proof of depth $\leq d-1$. Using the induction hypothesis we obtain $\mathsf{KT}^{\mathcal{S},2} \vdash \Diamond B, \Diamond\Sigma \mid B, \Gamma$ with a proof of depth $\leq d-1$, and with an application of $(\Diamond, \text{new})$ follows $\mathsf{KT}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Diamond B, \Gamma$ with a proof of depth $\leq d$.
2. The last step is $\frac{\Diamond\Sigma \mid B, \Diamond B, \Gamma}{\Diamond\Sigma \mid \Diamond B, \Diamond B, \Gamma}(\Diamond, \text{dup})$, where $A \equiv \Diamond B$: Let d be the depth of the proof of $\Diamond\Sigma \mid B, \Diamond B, \Gamma$. Since $(\Diamond, \text{dup})$ is strongly invertible we know that $\mathsf{KT}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid B, B, \Gamma$ with a proof of depth $\leq d-1$. Using the induction hypothesis we obtain $\mathsf{KT}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid B, \Gamma$ with a proof of depth $\leq d-1$, and with an application of $(\Diamond, \text{dup})$ follows $\mathsf{KT}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Diamond B, \Gamma$ with a proof of depth $\leq d$.

5.4.15 THEOREM equivalence of $\mathsf{KT}^{\mathcal{S}}$ and $\mathsf{KT}^{\mathcal{S},2}$

$$\mathsf{KT}^{\mathcal{S}} \vdash \epsilon \mid A \quad \Leftrightarrow \quad \mathsf{KT}^{\mathcal{S},2} \vdash \epsilon \mid A$$

Proof

We can reuse the proof of the corresponding theorem for $\mathsf{K}^{\mathcal{S}}$ and $\mathsf{K}^{\mathcal{S},2}$.

5.4.16 THEOREM $\mathsf{KT}^{\mathcal{S},2}$: termination

Backward proof search in $\mathsf{KT}^{\mathcal{S},2}$ for a sequent of the form $\epsilon \mid A$ always terminates.

Proof

We define $m(\Diamond\Sigma \mid \Gamma) := c^2 \cdot n_\Box(\Diamond\Sigma \mid \Gamma) + \text{length}(\Gamma)$, where $c := \text{length}(A)$ and $n_\Box(\Diamond\Sigma \mid \Gamma)$ is the number of $\Box$ symbols in the sequent $\Diamond\Sigma \mid \Gamma$ (e.g. $n_\Box(\Diamond\Box p_0 \mid \Box p_1 \vee \Box\Box p_0, \Diamond\Box p_0, \Diamond\Box p_0) = 6$).

The value computed by m decreases with every backward application of a rule of $\mathsf{KT}^{\mathcal{S},2}$:

1. $m(\Diamond\Sigma \mid A \vee B, \Gamma) > m(\Diamond\Sigma \mid A, B, \Gamma)$

2. $m(\Diamond\Sigma \mid A \land B, \Gamma) > m(\Diamond\Sigma \mid A, \Gamma)$, $m(\Diamond\Sigma \mid A \land B, \Gamma) > m(\Diamond\Sigma \mid B, \Gamma)$.
3. $m(\Diamond\Sigma \mid \Diamond A, \Gamma) > m(\Diamond A, \Diamond\Sigma \mid A, \Gamma)$
4. $m(\Diamond\Sigma \mid \Diamond A, \Gamma) > m(\Diamond\Sigma \mid A, \Gamma)$
5. Only the $(\Box)$ rule is interesting:

   $$\begin{aligned} & m(\Diamond\Sigma \mid \Box A, \Gamma) - m(\epsilon \mid A, \Sigma) \\ & \quad \geq c^2 \cdot 1 + \mathrm{length}(\Box A, \Gamma) - \mathrm{length}(A, \Sigma) \\ & \quad = c^2 + 1 + \mathrm{length}(\Gamma) - \mathrm{length}(\Sigma) \\ & \quad \geq c^2 + 1 - \mathrm{length}(\Sigma) \\ & \quad \geq 1. \end{aligned}$$

   The last inequality is correct since Σ contains only elements from $\mathrm{subfmls}(A)$, Σ contains no duplicate formulas, and $\mathrm{length}(\mathrm{subfmls}(A)) \leq (\mathrm{length}(A))^2$. Theorem 6.4.4 shows how important it is to remove duplicate $\Diamond$ formulas.

Example

Backward proof search for $\Diamond(p_0 \land p_2) \lor \neg p_0$ in $\mathsf{KT}^{\mathcal{S},2}$ terminates (cp. the example in remark 5.4.9).

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Diamond(p_0 \land p_2) \mid p_0, \neg p_0}\,(\mathrm{id}) \qquad \Diamond(p_0 \land p_2) \mid p_2, \neg p_0}{\Diamond(p_0 \land p_2) \mid p_0 \land p_2, \neg p_0}\,(\land)}{\epsilon \mid \Diamond(p_0 \land p_2), \neg p_0}\,(\Diamond, \mathrm{new})}{\epsilon \mid \Diamond(p_0 \land p_2) \lor \neg p_0}\,(\lor)$$

5.4.17 THEOREM $\mathsf{KT}^{\mathcal{S},2}$: extend (id)

If A is in negation normal form, then:

$$\mathsf{KT}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, \mathrm{nnf}(\neg A), \Gamma$$

Proof

We make an induction on $\mathrm{length}(A)$. The proof is analogous to the corresponding proof for $\mathsf{K}^{\mathcal{S},2}$ (theorem 5.2.16). We only show the case where $A \equiv \Box B$.

1. If $\mathrm{nnf}(\neg B) \notin \Sigma$, then we obtain $\mathsf{KT}^{\mathcal{S},2} \vdash \epsilon \mid B, \mathrm{nnf}(\neg B), \Sigma$ with the induction hypothesis, with an application of $(\Box)$ follows $\mathsf{KT}^{\mathcal{S},2} \vdash \Diamond\mathrm{nnf}(\neg B), \Diamond\Sigma \mid \Box B, \Gamma$, and with a application of $(\Diamond, \mathrm{new})$ finally $\mathsf{KT}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Diamond\mathrm{nnf}(\neg B), \Box B, \Gamma$.

   If $\mathrm{nnf}(\neg B) \in \Sigma$, then we obtain $\mathsf{KT}^{\mathcal{S},2} \vdash \epsilon \mid B, \Sigma$ with the induction hypothesis, and with an application of $(\Box)$ follows $\mathsf{KT}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Box B, \Diamond\mathrm{nnf}(\neg B), \Gamma$.

5.4.18 THEOREM $\mathsf{KT}^{\mathcal{S},2}$: use-check

Assume that we do backward proof search for a sequent of the form $\epsilon \mid D$. Let $\Diamond\Sigma \mid A, \Gamma$ be a sequent we obtain during this search, and let $\mathcal{P}$ be a proof in $\mathsf{KT}^{\mathcal{S},2}$ of this sequent. We mark it as follows:

- The main formulas in the instances of axioms are marked.
- If a side formula is marked, then the main formula is marked.
- If in the premise of an instance of $(\lor)$, $(\land)$, $(\Diamond, \mathrm{new})$ or $(\Diamond, \mathrm{dup})$ a formula in Γ is marked, then the corresponding formula in the multiset Γ in the conclusion is marked.

- If in the premise of an instance of $(\vee)$, $(\wedge)$, $(\Diamond, \text{new})$ or $(\Diamond, \text{dup})$ a formula in $\Diamond\Sigma$ is marked, then the corresponding formula in the multiset $\Diamond\Sigma$ in the conclusion is marked.
- If in the premise of an instance of $(\Box)$ a formula in Σ is marked, then the corresponding formula in the multiset $\Diamond\Sigma$ in the conclusion is marked.

If the formula A in the sequent $\Diamond\Sigma \mid A, \Gamma$ is not marked, then $\mathsf{KT}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$.

Proof

In the same way as in the proof of theorem 5.2.17 we prove the following: If $\mathcal{P}$ is a proof of $\Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$ in $\mathsf{KT}^{\mathcal{S},2}$ and no formula in $\Diamond\Delta$ or Π is marked, then $\mathsf{KT}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$.

The only additional case is the case where the last step in $\mathcal{P}$ is $\frac{\epsilon \mid B, \Delta, \Sigma}{\Diamond\Delta, \Diamond\Sigma \mid \Box B, \Pi_1, \Gamma}$ where $\Pi = \Box B, \Pi_1$. With the induction hypothesis we obtain $\mathsf{KT}^{\mathcal{S},2} \vdash \epsilon \mid \Sigma$.

First we consider two special cases. Assume that there is a $\Box$ formula $\Box C$ in the multiset Γ. With weakening we obtain $\mathsf{KT}^{\mathcal{S},2} \vdash \epsilon \mid \Sigma, C$, and with a $(\Box)$ application follows $\mathsf{KT}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$. If $\epsilon \mid \Gamma$ is provable, then we obtain the same result directly with weakening.

Now let S_1 be the sequent $\Diamond\Delta, \Diamond\Sigma \mid \Box B, \Pi_1, \Gamma$. We go from S_1 towards the root of $\mathcal{P}$, i.e. towards $\epsilon \mid D$. Let S be the first sequent with an empty left hand side we find. We will always find such a sequent since the root is of this form (cp. remark 5.4.19). Thus for every $\Diamond C \in \Diamond\Sigma$ the formula $\Diamond C$ is the main formula of a $(\Diamond, \text{new})$ application between S_1 and S. Consequently for every $\Diamond C \in \Diamond\Sigma$ the formula C is an element of Γ or the main formula of an application of $(\vee)$, $(\wedge)$, $(\Diamond, \text{new})$ or $(\Diamond, \text{dup})$ between S_1 and S. (This is not true in the case of our sequent calculi for K.) Starting from $\epsilon \mid \Sigma$ we apply $(\wedge)$, $(\vee)$, $(\Diamond, \text{new})$, $(\Diamond, \text{dup})$ backwards. We do it in the same way as on the branch between S and S_1 and always choose the same premise in order to continue. We obtain a branch that begins with $\epsilon \mid \Sigma$ and ends in a sequent of the form $\Diamond\Sigma' \mid \Gamma'$, where $\Gamma' \subseteq \Gamma$ and $\Sigma' \subseteq \Sigma$. Since $(\wedge)$, $(\vee)$, $(\Diamond, \text{new})$, $(\Diamond, \text{dup})$ are invertible and $\epsilon \mid \Sigma$ is provable, we know that $\Diamond\Sigma' \mid \Gamma'$ is provable. But then also $\Diamond\Sigma \mid \Gamma$ is provable because of weakening.

5.4.19 REMARK $\mathsf{KT}^{\mathcal{S},2}$: use-check

In theorem 5.4.18, we cannot omit the condition that we do backward proof search for a sequent with an empty left hand side.

Example

In the proof of the sequent $\Diamond(p_0 \vee \neg p_0) \mid \Box p_1$ no subformula of $\Box p_1$ is a main formula in an instance of an (id) axiom. Nevertheless $\mathsf{KT}^{\mathcal{S},2} \nvdash \Diamond(p_0 \vee \neg p_0) \mid p_2$, since no rule is applicable backwards. This cannot happen during backward proof search in $\mathsf{KT}^{\mathcal{S},2}$ for a sequent of the form $\epsilon \mid D$.

5.5 $\mathsf{KT} + T$

5.5.1 DEFINITION sequent calculus $(\mathsf{KT} + T)^{\mathcal{S}}$

$(\mathsf{KT} + T)^{\mathcal{S}}$ is the calculus $\mathsf{KT}^{\mathcal{S}}$ with the $(\Box)$ rule replaced by

$$\frac{\epsilon \mid A, \Sigma, \text{nnf}(\neg T)}{\Diamond\Sigma \mid \Box A, \Gamma}\ (\Box)$$

The main formula of this rule is $\Box A$.

Example

If $T = p_0, \Box p_1$, then $(\mathsf{KT} + T)^{\mathcal{S}} \vdash \epsilon \mid \Box(p_0 \wedge p_1), \text{nnf}(\neg T)$, as the following proof shows.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Diamond\neg p_1 \mid p_0, \neg p_1, \neg p_0, \neg p_1}\,(\mathrm{id}) \quad \dfrac{}{\Diamond\neg p_1 \mid p_1, \neg p_1, \neg p_0, \neg p_1}\,(\mathrm{id})}{\Diamond\neg p_1 \mid p_0 \wedge p_1, \neg p_1, \neg p_0, \neg p_1}\,(\wedge)}{\epsilon \mid p_0 \wedge p_1, \neg p_1, \neg p_0, \Diamond\neg p_1}\,(\Diamond)}{\Diamond\neg p_1 \mid \Box(p_0 \wedge p_1), \neg p_0, \neg p_1}\,(\Box)}{\epsilon \mid \Box(p_0 \wedge p_1), \neg p_0, \Diamond\neg p_1}\,(\Diamond)$$

5.5.2 THEOREM $(\mathsf{KT} + T)^{\mathcal{S}}$: weakening

If $(\mathsf{KT} + T)^{\mathcal{S}} \vdash \Diamond\Sigma \mid \Gamma$, then $(\mathsf{KT} + T)^{\mathcal{S}} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$. Moreover, if d is the depth of the proof of $\Diamond\Sigma \mid \Gamma$, then there exists a proof of $\Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$ whose depth is at most d.

Proof

Except for the multiset $\mathrm{nnf}(\neg T)$ in the premise of $(\Box)$ applications the proof is the same as the corresponding proof for $\mathsf{KT}^{\mathcal{S}}$ (theorem 5.4.3).

5.5.3 THEOREM $(\mathsf{KT} + T)^{\mathcal{S}}$: invertible rules

The rule $(\Box)$ of $(\mathsf{KT} + T)^{\mathcal{S}}$ is not invertible. All the other rules are strongly invertible.

Proof

Analogous to the corresponding proof for $\mathsf{KT}^{\mathcal{S}}$ (theorem 5.4.4).

5.5.4 THEOREM $(\mathsf{KT} + T)^{\mathcal{S}}$: duplicate formulas

$$\begin{array}{rcl} (\mathsf{KT} + T)^{\mathcal{S}} \vdash \Diamond\Sigma \mid A, A, \Gamma & \Rightarrow & (\mathsf{KT} + T)^{\mathcal{S}} \vdash \Diamond\Sigma \mid A, \Gamma \\ (\mathsf{KT} + T)^{\mathcal{S}} \vdash \Diamond A, \Diamond A, \Diamond\Sigma \mid \Gamma & \Rightarrow & (\mathsf{KT} + T)^{\mathcal{S}} \vdash \Diamond A, \Diamond\Sigma \mid \Gamma \end{array}$$

Proof

The only difference between this proof and the corresponding proof for $\mathsf{KT}^{\mathcal{S}}$ (theorem 5.4.5) is the multiset $\mathrm{nnf}(\neg T)$ in the premises of $(\Box)$ application.

5.5.5 THEOREM equivalence of $(\mathsf{KT} + T)^{\mathcal{G},2}$ and $(\mathsf{KT} + T)^{\mathcal{S}}$

If A is in negation normal form, then:

$$(\mathsf{KT} + T)^{\mathcal{G},2} \vdash \boxed{\boxed{\epsilon \mid A, \mathrm{nnf}(\neg T)}} \quad \Leftrightarrow \quad (\mathsf{KT} + T)^{\mathcal{S}} \vdash A, \mathrm{nnf}(\neg T)$$

Proof

The only difference between this proof and the corresponding proof for $\mathsf{KT}^{\mathcal{S}}$ (theorem 5.4.6) is the multiset $\mathrm{nnf}(\neg T)$ in the premises of $(\Box)$ applications. Since we have $\mathrm{nnf}(\neg T)$ both in the $(\Box)$ rule of $(\mathsf{KT} + T)^{\mathcal{G},2}$ and in the $(\Box)$ rule of $(\mathsf{KT} + T)^{\mathcal{S}}$, the translation at the end of the proof causes no difficulties.

5.5.6 THEOREM proofs in $(\mathsf{KT} + T)^{\mathcal{H}}$ from proofs in $(\mathsf{KT} + T)^{\mathcal{S}}$

$$(\mathsf{KT} + T)^{\mathcal{S}} \vdash \mathrm{nnf}(A), \mathrm{nnf}(\neg T) \quad \Rightarrow \quad (\mathsf{KT} + T)^{\mathcal{H}} \vdash A$$

Proof

We can reuse most parts of the corresponding proof for $(\mathsf{KT}+T)^{\mathcal{H}}$ and $(\mathsf{KT}+T)^{\mathcal{S}}$ (theorem 5.4.7). In addition, we have to show that if $T = C_1, \ldots, C_n$, then $(\mathsf{KT}+T)^{\mathcal{H}} \vdash B \vee C_1 \vee \ldots \vee C_n$ implies $(\mathsf{KT}+T)^{\mathcal{H}} \vdash B$. This follows from $(\mathsf{KT}+T)^{\mathcal{S}} \vdash C_i$ (for all $i \in \{1, \ldots, n\}$) with applications of (cpc) and (mp).

5.5.7 DEFINITION sequent calculus $(\mathsf{KT}+T)^{\mathcal{S},2}$

$(\mathsf{KT}+T)^{\mathcal{S},2}$ is the calculus $\mathsf{KT}^{\mathcal{S},2}$ with the $(\Box)$ rule replaced by

$$\frac{\epsilon \mid A, \Sigma, \mathrm{nnf}(\neg T)}{\Diamond\Sigma \mid \Box A, \Gamma}\ (\Box)$$

The main formula of this rule is $\Box A$.

5.5.8 THEOREM $(\mathsf{KT}+T)^{\mathcal{S},2}$: weakening

If $(\mathsf{KT}+T)^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$, then $(\mathsf{KT}+T)^{\mathcal{S},2} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$. Moreover, if d is the depth of the proof of $\Diamond\Sigma \mid \Gamma$, then there exists a proof of $\Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$ whose depth is at most d.

Proof

Except for the multiset $\mathrm{nnf}(\neg T)$ in the premise of $(\Box)$ applications the proof is the same as the corresponding proof for $\mathsf{KT}^{\mathcal{S},2}$ (theorem 5.4.12).

5.5.9 THEOREM $(\mathsf{KT}+T)^{\mathcal{S},2}$: invertible rules

The rule $(\Box)$ of $(\mathsf{KT}+T)^{\mathcal{S},2}$ is not invertible. All the other rules are strongly invertible.

Proof

Analogous to the corresponding proof for $\mathsf{KT}^{\mathcal{S},2}$ (theorem 5.4.13).

5.5.10 THEOREM $(\mathsf{KT}+T)^{\mathcal{S},2}$: duplicate formulas

$$(\mathsf{KT}+T)^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, A, \Gamma \quad \Rightarrow \quad (\mathsf{KT}+T)^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, \Gamma$$

Proof

The only difference between this proof and the corresponding proof for $\mathsf{KT}^{\mathcal{S},2}$ (theorem 5.4.14) is the multiset $\mathrm{nnf}(\neg T)$ in the premises of $(\Box)$ applications.

5.5.11 THEOREM equivalence of $(\mathsf{KT}+T)^{\mathcal{S}}$ and $(\mathsf{KT}+T)^{\mathcal{S},2}$

$$(\mathsf{KT}+T)^{\mathcal{S}} \vdash \epsilon \mid A, \mathrm{nnf}(\neg T) \quad \Leftrightarrow \quad (\mathsf{KT}+T)^{\mathcal{S},2} \vdash \epsilon \mid A, \mathrm{nnf}(\neg T)$$

Proof

Analogous to the corresponding proof for $\mathsf{KT}^{\mathcal{S}}$ and $\mathsf{KT}^{\mathcal{S},2}$ (theorem 5.4.6).

5.5.12 REMARK $(\mathsf{KT}+T)^{\mathcal{S},2}$: non-termination

Backward proof search in $(\mathsf{KT}+T)^{\mathcal{S},2}$ does in general not terminate.

Example

If $T = p_0 \vee \Diamond p_1$, then backward proof search for $\Diamond p_0, \mathrm{nnf}(\neg T)$ in $(\mathsf{KT}+T)^{\mathcal{S},2}$ does not terminate if we use a depth-first search on the rightmost branch.

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{(id)}
\UnaryInfC{$\Diamond p_0 \mid p_0, \neg p_0$}
\AxiomC{}
\RightLabel{(id)}
\UnaryInfC{$\epsilon \mid \neg p_1, p_0, \neg p_0$}
\AxiomC{$\epsilon \mid \neg p_1, \neg p_0$}
\AxiomC{$\vdots$}
\RightLabel{$(\wedge)$}
\UnaryInfC{$\epsilon \mid \neg p_1, \neg p_0 \wedge \Box \neg p_1$}
\RightLabel{$(\Box)$}
\UnaryInfC{$\epsilon \mid \neg p_1, \Box \neg p_1$}
\RightLabel{$(\wedge)$}
\BinaryInfC{$\epsilon \mid \neg p_1, \neg p_0 \wedge \Box \neg p_1$}
\RightLabel{$(\Box)$}
\UnaryInfC{$\epsilon \mid \neg p_1, p_0, \Box \neg p_1$}
\RightLabel{$(\wedge)$}
\BinaryInfC{$\epsilon \mid \neg p_1, p_0, \neg p_0 \wedge \Box \neg p_1$}
\RightLabel{$(\Box)$}
\UnaryInfC{$\Diamond p_0 \mid p_0, \Box \neg p_1$}
\RightLabel{$(\wedge)$}
\BinaryInfC{$\Diamond p_0 \mid p_0, \neg p_0 \wedge \Box \neg p_1$}
\RightLabel{$(\Diamond, \text{new})$}
\UnaryInfC{$\epsilon \mid \Diamond p_0, \neg p_0 \wedge \Box \neg p_1$}
\end{prooftree}
$$

5.5.13 REMARK from $(\mathsf{KT}+T)^{\mathcal{S},2}$ to $(\mathsf{KT}+T)^{\mathcal{S},3}$

We use the same idea as when going from $(\mathsf{K}+T)^{\mathcal{S},2}$ to $(\mathsf{K}+T)^{\mathcal{S},3}$. We add a history to the sequents and use a loop-check in the $(\Box)$ rule to ensure termination. With the conditions in the rules $(\Diamond, \text{new})$ and $(\Diamond, \text{dup})$ we limit the number of possible elements of the history.

5.5.14 DEFINITION sequent calculus $(\mathsf{KT}+T)^{\mathcal{S},3}$

axioms:

$$\frac{}{H \mid \Diamond\Sigma \mid \mathsf{true}, \Gamma}\ (\mathsf{true}) \qquad\qquad \frac{}{H \mid \Diamond\Sigma \mid P, \neg P, \Gamma}\ (\text{id})$$

rules:

$$\frac{H \mid \Diamond\Sigma \mid A, B, \Gamma}{H \mid \Diamond\Sigma \mid A \vee B, \Gamma}\ (\vee) \qquad\qquad \frac{H \mid \Diamond\Sigma \mid A, \Gamma \quad H \mid \Diamond\Sigma \mid B, \Gamma}{H \mid \Diamond\Sigma \mid A \wedge B, \Gamma}\ (\wedge)$$

$$\frac{H \mid \Diamond A, \Diamond\Sigma \mid A, \Gamma}{H \mid \Diamond\Sigma \mid \Diamond A, \Gamma}\ \Diamond A \notin \Diamond\Sigma \quad (\Diamond, \text{new}) \qquad\qquad \frac{H \mid \Diamond\Sigma \mid A, \Gamma}{H \mid \Diamond\Sigma \mid \Diamond A, \Gamma}\ \Diamond A \in \Diamond\Sigma \quad (\Diamond, \text{dup})$$

$$\frac{[\Box A, \Diamond\Sigma], H \mid \epsilon \mid A, \Sigma, \mathrm{nnf}(\neg T)}{H \mid \Diamond\Sigma \mid \Box A, \Gamma}\ [\Box A, \Diamond\Sigma] \notin H \quad (\Box)$$

main formulas: true in (true), P and $\neg P$ in (id), $A \vee B$ in $(\vee)$, $A \wedge B$ in $(\wedge)$, $\Diamond A$ in $(\Diamond, \text{new})$ and $(\Diamond, \text{dup})$, $\Box A$ in $(\Box)$

side formulas: A and B in $(\vee)$, A and B in $(\wedge)$, A and $\Diamond A$ in $(\Diamond, \text{new})$, A in $(\Diamond, \text{dup})$, A in $(\Box)$

5.5.15 THEOREM equivalence of $(\mathsf{KT}+T)^{\mathcal{S},2}$ and $(\mathsf{KT}+T)^{\mathcal{S},3}$

$$(\mathsf{KT}+T)^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma, \mathrm{nnf}(\neg T) \quad \Leftrightarrow \quad (\mathsf{KT}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma, \mathrm{nnf}(\neg T)$$

Proof

'$\Leftarrow$':

Let $\mathcal{P}$ be a proof of $\epsilon \mid \Diamond\Sigma \mid \Gamma, \mathrm{nnf}(\neg T)$ in $(\mathsf{KT}+T)^{\mathcal{S},3}$. We omit the history in $\mathcal{P}$. With an induction on proof depth we can easily prove that we obtain a proof of $\Diamond\Sigma \mid \Gamma, \mathrm{nnf}(\neg T)$ in $(\mathsf{KT}+T)^{\mathcal{S},2}$.

'⇒':

We use the same transformation as in the corresponding proof for $(\mathsf{K}+T)^{\mathcal{S},2}$ and $(\mathsf{K}+T)^{\mathcal{S},3}$ (theorem 5.3.15) in order to obtain a proof in which no sequent occurs twice on a branch.

Now we add the history (beginning at the root of the proof) to the sequents. Again we proceed as in the proof of theorem 5.3.15. We show only the cases where one of $(\Box)$, $(\Diamond,\text{new})$, $(\Diamond,\text{dup})$ is involved.

1. $\frac{\epsilon|A,\Sigma,\text{nnf}(\neg T)}{\Diamond\Sigma|\Box A,\Gamma}(\Box)$ is a step in the proof, and the sequent $\Diamond\Sigma \mid \Box A,\Gamma$ was transformed into $H \mid \Diamond\Sigma \mid \Box A,\Gamma$: Then $\epsilon \mid A,\Sigma,\text{nnf}(\neg T)$ is transformed into $[\Box A,\Diamond\Sigma],H \mid \Diamond\Sigma \mid \Box A,\Gamma$. Because of the transformation we know that between the root of the new proof and this $(\Box)$ application there is no $(\Box)$ application with a premise of the form $H' \mid \epsilon \mid A,\Sigma,\text{nnf}(\neg T)$. Thus $[\Box A,\Diamond\Sigma] \notin H$.
2. $\frac{\Diamond A,\Diamond\Sigma|A,\Gamma}{\Diamond\Sigma|\Diamond A,\Gamma}(\Diamond,\text{new})$ is a step in the proof, and the sequent $\Diamond\Sigma \mid \Diamond A,\Gamma$ was transformed into $H \mid \Diamond\Sigma \mid \Diamond A,\Gamma$: Then $\Diamond A,\Diamond\Sigma \mid A,\Gamma$ is transformed into $H \mid \Diamond A,\Diamond\Sigma \mid A,\Gamma$.
3. $\frac{\Diamond\Sigma|A,\Gamma}{\Diamond\Sigma|\Diamond A,\Gamma}(\Diamond,\text{dup})$ is a step in the proof, and the sequent $\Diamond\Sigma \mid \Diamond A,\Gamma$ was transformed into $H \mid \Diamond\Sigma \mid \Diamond A,\Gamma$: Then $\Diamond\Sigma \mid A,\Gamma$ is transformed into $H \mid \Diamond\Sigma \mid A,\Gamma$.

5.5.16 THEOREM $(\mathsf{KT}+T)^{\mathcal{S},3}$: termination

Backward proof search in $(\mathsf{KT}+T)^{\mathcal{S},3}$ for sequents of the form $\epsilon \mid \epsilon \mid A,\text{nnf}(\neg T)$ always terminates.

Proof

Assume that we do backward proof search for $\epsilon \mid \epsilon \mid A,\text{nnf}(\neg T)$. We define $m(H \mid \Diamond\Sigma \mid \Gamma) := (c+c^2)\cdot(2^c - \text{card}(H)) + \text{length}(\Gamma)$, where $c := \text{length}(A) + \text{length}(\text{nnf}(\neg T))$.

If we do backward proof search for $\epsilon \mid \epsilon \mid A$, then certainly $m(H \mid \Diamond\Sigma \mid \Gamma) \geq 0$ for all sequents $H \mid \Diamond\Sigma \mid \Gamma$ that can occur during the search, because of the subformula property of $(\mathsf{KT}+T)^{\mathcal{S},3}$ and since neither H nor $\Diamond\Sigma$ contain duplicate elements.

Obviously the value computed by m decreases when one of the rules $(\wedge)$, $(\vee)$, $(\Diamond,\text{new})$, $(\Diamond,\text{dup})$ is applied backwards. Since there are no duplicate elements in $\Diamond\Sigma$ this is also the case for the $(\Box)$ rule:

$$\begin{aligned}
&m(H \mid \Diamond\Sigma \mid \Box A,\Gamma) - m(H,[\Box A,\Diamond\Sigma] \mid \epsilon \mid A,\Sigma,\text{nnf}(\neg T))\\
&\quad = (c+c^2)\cdot(\text{card}(H,[\Box A,\Diamond\Sigma]) - \text{card}(H)) + \text{length}(\Box A,\Gamma) - \text{length}(A,\Sigma,\text{nnf}(\neg T))\\
&\quad = (c+c^2) + 1 + \text{length}(\Gamma) - \text{length}(\Sigma) - \text{length}(\text{nnf}(\neg T))\\
&\quad \geq (c+c^2) + 1 - \text{length}(\Sigma) - \text{length}(\text{nnf}(\neg T))\\
&\quad \geq (c+c^2) + 1 - c^2 - c\\
&\quad = 1.
\end{aligned}$$

5.5.17 THEOREM proofs in $(\mathsf{KT}+T)^{\mathcal{S},3}$

If $(\mathsf{KT}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma,\text{nnf}(\neg T)$, then there exists a proof of $\epsilon \mid \Diamond\Sigma \mid \Gamma,\text{nnf}(\neg T)$ in $(\mathsf{KT}+T)^{\mathcal{S},3}$ such that $(\Box)$ is only applied backwards if no other rule is applicable backwards.

Proof

Because of theorem 5.5.9 and theorem 5.5.15 there exists a proof of $\Diamond\Sigma \mid \Gamma,\text{nnf}(\neg T)$ in $(\mathsf{KT}+T)^{\mathcal{S},2}$ such that $(\Box)$ is only applied backwards if no other rule is applicable backwards. We now use the construction from the proof of theorem 5.5.15 to transform it into a proof of $\epsilon \mid \Diamond\Sigma \mid \Gamma,\text{nnf}(\neg T)$ in $(\mathsf{KT}+T)^{\mathcal{S},3}$ of the desired form.

5.5.18 THEOREM $(\mathsf{KT} + T)^{\mathcal{S},3}$: permutation of rules

For the calculus $(\mathsf{KT} + T)^{\mathcal{S},3}$, the following applies for all axioms x and all rules r:

- If r is not the rule $(\Box)$, then $x \leadsto rx$.
- If neither r nor s is the rule $(\Box)$, then $rs \cdots \leadsto sr \cdots$.

Proof

Since the only difference between $(\mathsf{KT} + T)^{\mathcal{S},2}$ and $(\mathsf{KT} + T)^{\mathcal{S},3}$ is in the $(\Box)$ rule, we can use the same transformations as in the proof of theorem 5.5.9.

5.5.19 REMARK invertible rules in $(\mathsf{KT} + T)^{\mathcal{S},3}$

We saw in remark 5.3.19 that the rule $(\Diamond, \text{new})$ of $(\mathsf{K} + T)^{\mathcal{S},3}$ is not invertible. Also the rule $(\Diamond, \text{new})$ of $(\mathsf{KT} + T)^{\mathcal{S},3}$ is not invertible, since

$$\dfrac{\dfrac{}{[\Box\mathsf{true}], [\Box\mathsf{true}, \Diamond p_1] \mid \epsilon \mid \mathsf{true}}\,(\mathsf{true})}{[\Box\mathsf{true}, \Diamond p_1] \mid \epsilon \mid \Box\mathsf{true}, \Diamond p_1}\,(\Box)$$

is a proof of $[\Box\mathsf{true}, \Diamond p_1] \mid \epsilon \mid \Box\mathsf{true}, \Diamond p_1$ in $(\mathsf{KT} + T)^{\mathcal{S},3}$, but $(\mathsf{KT} + T)^{\mathcal{S},3} \nvdash [\Box\mathsf{true}, \Diamond p_1] \mid \Diamond p_1 \mid \Box\mathsf{true}, p_1$.

The two theorems 5.5.17 and 5.5.18 show that if we do backward proof search in $(\mathsf{KT} + T)^{\mathcal{S},3}$ for a sequent of the form $\epsilon \mid \epsilon \mid A, \text{nnf}(\neg T)$, then we can do this as if the rules $(\vee)$, $(\wedge)$, $(\Diamond, \text{new})$, $(\Diamond, \text{dup})$ were invertible.

5.5.20 THEOREM $(\mathsf{KT} + T)^{\mathcal{S},3}$: empty the history

$$(\mathsf{KT} + T)^{\mathcal{S},3} \vdash H \mid \Diamond\Sigma \mid \Gamma \Rightarrow (\mathsf{KT} + T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma$$

Proof

We proceed as in the corresponding proof for $(\mathsf{K} + T)^{\mathcal{S},3}$ (theorem 5.3.20).

5.5.21 THEOREM $(\mathsf{KT} + T)^{\mathcal{S},3}$: extend (id)

If A is in negation normal form, then:

$$(\mathsf{KT} + T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid A, \text{nnf}(\neg A), \Gamma$$

Proof

With an induction on length(A) we prove $(\mathsf{KT} + T)^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, \text{nnf}(\neg A), \Gamma$. We show only the case where $A \equiv \Box B$.

1. $A \equiv \Box B$: If $\Diamond\text{nnf}(\neg B) \notin \Diamond\Sigma$, then we have $(\mathsf{KT} + T)^{\mathcal{S},2} \vdash \epsilon \mid B, \text{nnf}(\neg B), \Sigma, \text{nnf}(\neg T)$ because of the induction hypothesis. With a $(\Box)$ application we obtain $(\mathsf{KT} + T)^{\mathcal{S},2} \vdash \Diamond\text{nnf}(\neg B), \Diamond\Sigma \mid \Box B, \text{nnf}(\neg B), \Gamma$, and with a $(\Diamond, \text{new})$ application $(\mathsf{KT} + T)^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Box B, \Diamond\text{nnf}(\neg B), \Gamma$. If $\Diamond\text{nnf}(\neg B) \in \Diamond\Sigma$, then we have $(\mathsf{KT} + T)^{\mathcal{S}} \vdash \epsilon \mid B, \Sigma, \text{nnf}(\neg T)$, thus with a $(\Box)$ application $(\mathsf{KT} + T)^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Box B, \Diamond\text{nnf}(\neg B), \Gamma$.

Now we use theorem 5.5.15 to obtain $(\mathsf{KT} + T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid A, \text{nnf}(\neg A), \Gamma$.

5.5.22 THEOREM $(\mathsf{KT}+T)^{\mathcal{S},3}$: duplicate formulas

$$(\mathsf{KT}+T)^{\mathcal{S},3} \vdash H \mid \Diamond\Sigma \mid A, A, \Gamma \quad \Rightarrow \quad (\mathsf{KT}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid A, \Gamma$$

Proof

We proceed as in the corresponding proof for $(\mathsf{K}+T)^{\mathcal{S},3}$ (theorem 5.3.22).

With theorem 5.5.20 we obtain $(\mathsf{KT}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid A, A, \Gamma$ and with theorem 5.5.15 follows $(\mathsf{KT}+T)^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, A, \Gamma$. Thus $(\mathsf{KT}+T)^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, \Gamma$ because of theorem 5.5.10. With theorem 5.5.15 we finally obtain $(\mathsf{KT}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid A, \Gamma$.

5.5.23 THEOREM $(\mathsf{KT}+T)^{\mathcal{S},3}$: use-check

Assume that we do backward proof search for a sequent of the form $\epsilon \mid \epsilon \mid D$. Let $H \mid \Diamond\Sigma \mid A, \Gamma$ be a sequent we obtain during this search, and let $\mathcal{P}$ be a proof in $(\mathsf{KT}+T)^{\mathcal{S},3}$ of this sequent. We mark it as follows:

- The main formulas in the instances of axioms are marked.
- If a side formula is marked, then the main formula is marked.
- If in the premise of an instance of $(\vee)$, $(\wedge)$, $(\Diamond, \text{new})$ or $(\Diamond, \text{dup})$ a formula in Γ is marked, then the corresponding formula in the multiset Γ in the conclusion is marked.
- If in the premise of an instance of $(\vee)$, $(\wedge)$, $(\Diamond, \text{new})$ or $(\Diamond, \text{dup})$ a formula in $\Diamond\Sigma$ is marked, then the corresponding formula in the multiset $\Diamond\Sigma$ in the conclusion is marked.
- If in the premise of an instance of $(\Box)$ a formula in Σ is marked, then the corresponding formula in the multiset $\Diamond\Sigma$ in the conclusion is marked.

If the formula A in the sequent $H \mid \Diamond\Sigma \mid A, \Gamma$ is not marked, then $(\mathsf{KT}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma$.

Proof

First we prove the theorem for the calculus $(\mathsf{KT}+T)^{\mathcal{S},2}$, defining marked proofs in $(\mathsf{KT}+T)^{\mathcal{S},2}$ exactly as described above for $(\mathsf{KT}+T)^{\mathcal{S},3}$. Except for the multiset $\text{nnf}(\neg T)$ in the premises of $(\Box)$ applications this is the same proof as the one of theorem 5.4.18.

Now we take the marked proof $\mathcal{P}$ of $H \mid \Diamond\Sigma \mid A, \Gamma$ and omit all histories, but do not change the marks. We call the result $\mathcal{P}'$. With an induction on proof depth we can easily show that $\mathcal{P}'$ is a marked proof of $\Diamond\Sigma \mid A, \Gamma$ in $(\mathsf{KT}+T)^{\mathcal{S},2}$. Thus A is not marked, and therefore $(\mathsf{KT}+T)^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$. With theorem 5.5.15 we obtain $(\mathsf{KT}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma$.

5.5.24 THEOREM (empty) is admissible

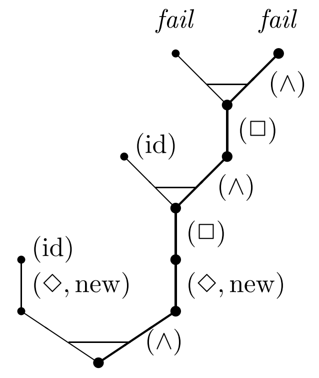
If $\epsilon \mid \Diamond\Sigma \mid \Gamma$ is provable in the calculus $(\mathsf{KT}+T)^{\mathcal{S},3}$ plus the rule $\frac{\epsilon|\Diamond\Sigma\Gamma}{H|\Diamond\Sigma\Gamma}$(empty), then $(\mathsf{KT}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma$.

Proof

Let $\mathcal{P}$ be a proof of $\epsilon \mid \Diamond\Sigma \mid \Gamma$ in $(\mathsf{KT}+T)^{\mathcal{S},3}$ plus the rule (empty). If we remove the applications of (empty) and all histories, then we obtain a proof of $\Diamond\Sigma \mid \Gamma$ in $(\mathsf{KT}+T)^{\mathcal{S},2}$, i.e. $(\mathsf{KT}+T)^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$. With theorem 5.5.15, $(\mathsf{KT}+T)^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma$.

5.5.25 REMARK using optimisations during proof search in $(\mathsf{KT}+T)^{\mathcal{S},3}$

The preceding theorem has the same purpose as the corresponding theorem for $(\mathsf{K}+T)^{\mathcal{S},3}$: It makes it possible to apply the theorems 5.5.21, 5.5.22, 5.5.23 during backward proof search even



Two branches of the search tree fail. Since no backtracking is possible in this example, we therefore have $(\mathsf{KT}+T)^{\mathcal{S},3} \vdash \epsilon \mid \epsilon \mid \Diamond p_0, \mathrm{nnf}(\neg T)$. The branch that fails because the right hand side of the sequent contains only literals corresponds to the countermodel without a loop. The branch (drawn in thick lines) that fails because of the loop check condition in the (□) rule corresponds to the countermodel with the loop.

Figure 5.j: The search tree of the example in remark 5.5.26. See figure 5.k for the corresponding non-proof and figure 5.l for the resulting countermodels.

if the history is not empty. The only condition is that we started with a sequent of the form $\epsilon \mid \epsilon \mid A$. The idea is the same as in the case of $\mathsf{K}^{\mathcal{S},3}$; therefore we just refer to remark 5.3.25.

5.5.26 REMARK $(\mathsf{KT}+T)^{\mathcal{S},3}$: backward proof search

We can extract countermodels from a failed backward proof search in the same way as in the case of $(\mathsf{K}+T)^{\mathcal{S},3}$.

Example

We take the theory $[p_0 \vee \Diamond p_1]$ and the formula $\Diamond p_0$ as in the example in remark 5.5.12.

See figure 5.j for the search tree and 5.k for the corresponding non-proof. Figure 5.l shows the two resulting countermodels.

5.6 S4

5.6.1 DEFINITION sequent calculus $\mathsf{S4}^{\mathcal{S}}$

axioms:

$$\frac{}{\Diamond\Sigma \mid \mathsf{true}, \Gamma}\ (\mathsf{true}) \qquad \frac{}{\Diamond\Sigma \mid P, \neg P, \Gamma}\ (\mathrm{id})$$

rules:

$$\frac{\Diamond\Sigma \mid A, B, \Gamma}{\Diamond\Sigma \mid A \vee B, \Gamma}\ (\vee) \qquad \frac{\Diamond\Sigma \mid A, \Gamma \quad \Diamond\Sigma \mid B, \Gamma}{\Diamond\Sigma \mid A \wedge B, \Gamma}\ (\wedge)$$

$$\frac{\Diamond A, \Diamond\Sigma \mid A, \Gamma}{\Diamond\Sigma \mid \Diamond A, \Gamma}\ (\Diamond) \qquad \frac{\epsilon \mid A, \Diamond\Sigma}{\Diamond\Sigma \mid \Box A, \Gamma}\ (\Box)$$

main formulas: true in (true), P and $\neg P$ in (id), $A \vee B$ in (∨), $A \wedge B$ in (∧), $\Diamond A$ in (◇), $\Box A$ in (□)

$$
\dfrac{
\dfrac{\dfrac{}{\epsilon \mid \Diamond p_0 \mid p_0, \neg p_0}\,(\text{id})}{\epsilon \mid \epsilon \mid \Diamond p_0, \neg p_0}\,(\Diamond, \text{new})
\qquad
\dfrac{\dfrac{\dfrac{
\dfrac{}{[\Box\neg p_1, \Diamond p_0] \mid \epsilon \mid \neg p_1, p_0, \neg p_0}\,(\text{id})
\quad
\dfrac{\dfrac{[\Box\neg p_1], [\Box\neg p_1, \Diamond p_0] \mid \epsilon \mid \neg p_1, \neg p_0 \quad [\Box\neg p_1], [\Box\neg p_1, \Diamond p_0] \mid \epsilon \mid \neg p_1, \Box\neg p_1}{[\Box\neg p_1], [\Box\neg p_1, \Diamond p_0] \mid \epsilon \mid \neg p_1, \neg p_0 \wedge \Box\neg p_1}\,(\wedge)}{[\Box\neg p_1, \Diamond p_0] \mid \epsilon \mid \neg p_1, p_0, \Box\neg p_1}\,(\Box)
}{[\Box\neg p_1, \Diamond p_0] \mid \epsilon \mid \neg p_1, p_0, \neg p_0 \wedge \Box\neg p_1}\,(\wedge)}{\epsilon \mid \Diamond p_0 \mid p_0, \Box\neg p_1}\,(\Box)}{\epsilon \mid \epsilon \mid \Diamond p_0, \Box\neg p_1}\,(\Diamond, \text{new})
}{\epsilon \mid \epsilon \mid \Diamond p_0, \neg p_0 \wedge \Box\neg p_1}\,(\wedge)
$$

Figure 5.k: The non-proof of the example in remark 5.5.26. See figure 5.j for the corresponding search tree and figure 5.l for the resulting countermodels.

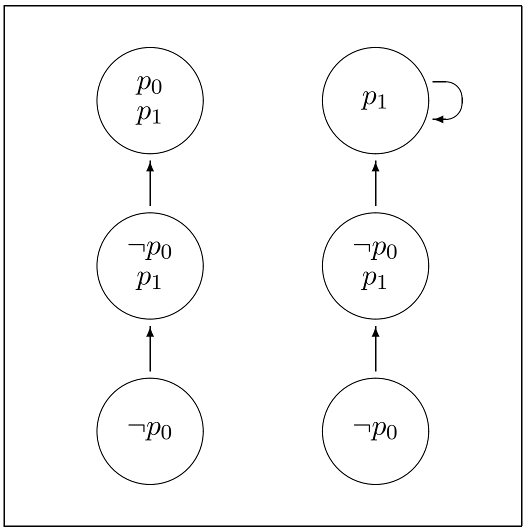


Figure 5.l: The two countermodels of the example in remark 5.5.26. See figure 5.j for the corresponding search tree.

Example

$\mathsf{S4}^{\mathcal{S}} \vdash \epsilon \mid \Diamond\neg p_1 \vee \Box\Box p_1$, as the following proof shows.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Diamond\neg p_1 \mid p_1, \neg p_1}\,(\mathrm{id})}{\epsilon \mid p_1, \Diamond\neg p_1}\,(\Diamond)}{\Diamond\neg p_1 \mid \Box p_1, \neg p_1}\,(\Box)}{\epsilon \mid \Box p_1, \Diamond\neg p_1}\,(\Diamond)}{\Diamond\neg p_1 \mid \neg p_1, \Box\Box p_1}\,(\Box)}{\epsilon \mid \Diamond\neg p_1, \Box\Box p_1}\,(\Diamond)}{\epsilon \mid \Diamond\neg p_1 \vee \Box\Box p_1}\,(\vee)$$

In contrast to $\mathsf{KT}^{\mathcal{S}}$, no $\Diamond$ is removed when $(\Box)$ is applied backwards. Therefore $\epsilon \mid \Diamond\neg p_1 \vee \Box\Box p_1$ is provable in $\mathsf{S4}^{\mathcal{S}}$, but not in $\mathsf{KT}^{\mathcal{S}}$.

5.6.2 THEOREM $\mathsf{S4}^{\mathcal{S}}$: weakening

If $\mathsf{S4}^{\mathcal{S}} \vdash \Diamond\Sigma \mid \Gamma$, then $\mathsf{S4}^{\mathcal{S}} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$. Moreover, if d is the depth of the proof of $\Diamond\Sigma \mid \Gamma$, then there exists a proof of $\Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$ whose depth is at most d.

Proof

The proof is analogous to the corresponding proof for $\mathsf{KT}^{\mathcal{S}}$ (theorem 5.4.3). We only show the case with $(\Box)$.

1. The last step is $\frac{\epsilon \mid A, \Diamond\Sigma}{\Diamond\Sigma \mid \Box A, \Gamma}$: With the induction hypothesis we obtain $\mathsf{S4}^{\mathcal{S}} \vdash \epsilon \mid A, \Diamond\Delta, \Diamond\Sigma$, and with an application of $(\Box)$ follows $\mathsf{S4}^{\mathcal{S}} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, \Box A, \Gamma$.

5.6.3 THEOREM $\mathsf{S4}^{\mathcal{S}}$: invertible rules

The rule $(\Box)$ of $\mathsf{S4}^{\mathcal{S}}$ is not invertible. All the other rules are strongly invertible.

Proof

Analogous to the proof of theorem 5.4.4. We only show $(\Box) \cdots \rightsquigarrow (\Diamond)(\Box) \cdots$. There we use the transformation below and weakening.

$$\frac{\epsilon \mid A, \Diamond\Sigma}{\Diamond\Sigma \mid \Box A, \Diamond B, \Gamma}(\Box) \quad \rightsquigarrow \quad \cfrac{\cfrac{\epsilon \mid A, \Diamond B, \Diamond\Sigma}{\Diamond B, \Diamond\Sigma \mid \Box A, B, \Gamma}(\Box)}{\Diamond\Sigma \mid \Box A, \Diamond B, \Gamma}(\Diamond)$$

5.6.4 THEOREM $\mathsf{S4}^{\mathcal{S}}$: duplicate formulas

$$\begin{aligned} \mathsf{S4}^{\mathcal{S}} \vdash \Diamond\Sigma \mid A, A, \Gamma \quad &\Rightarrow \quad \mathsf{S4}^{\mathcal{S}} \vdash \Diamond\Sigma \mid A, \Gamma \\ \mathsf{S4}^{\mathcal{S}} \vdash \Diamond A, \Diamond A, \Diamond\Sigma \mid \Gamma \quad &\Rightarrow \quad \mathsf{S4}^{\mathcal{S}} \vdash \Diamond A, \Diamond\Sigma \mid \Gamma \end{aligned}$$

Proof

The proof for $\mathsf{S4}^{\mathcal{S}}$ is analogous to the corresponding proof for $\mathsf{KT}^{\mathcal{S}}$ (theorem 5.4.5). We show only the cases where the last step is a $(\Box)$ application.

1. The last step is $\frac{\epsilon|B,\Diamond\Sigma}{\Diamond\Sigma|A,A,\Box B,\Gamma}(\Box)$: With an application of $(\Box)$ we obtain $\mathsf{S4}^{\mathcal{S}} \vdash \Diamond\Sigma \mid A, \Box B, \Gamma$.
2. The last step is $\frac{\epsilon|B,\Diamond C,\Diamond C,\Diamond\Sigma}{\Diamond C,\Diamond C,\Diamond\Sigma|\Box B,\Gamma}(\Box)$, where $A \equiv \Diamond C$: With the induction hypothesis we obtain $\mathsf{S4}^{\mathcal{S}} \vdash \epsilon \mid B, \Diamond C, \Diamond\Sigma$ and with an application of $(\Box)$ follows $\mathsf{S4}^{\mathcal{S}} \vdash \Diamond C, \Diamond\Sigma \mid \Box B, \Gamma$.
3. The last step is $\frac{\epsilon|B,\Diamond\Sigma}{\Diamond\Sigma|\Box B,\Box B,\Gamma}(\Box)$, where $A \equiv \Box B$: With an application of $(\Box)$ we obtain $\mathsf{S4}^{\mathcal{S}} \vdash \Diamond\Sigma \mid \Box B, \Gamma$.

5.6.5 THEOREM equivalence of $\mathsf{S4}^{\mathcal{G},2}$ and $\mathsf{S4}^{\mathcal{S}}$

If A is in negation normal form, then:

$$\mathsf{S4}^{\mathcal{G},2} \vdash \boxed{\boxed{\epsilon \mid A}} \quad \Leftrightarrow \quad \mathsf{S4}^{\mathcal{S}} \vdash \epsilon \mid A$$

Proof

‘$\Rightarrow$’:

We can proceed exactly as in the corresponding proof of the equivalence of $\mathsf{K}^{\mathcal{G},2}$ and $\mathsf{K}^{\mathcal{S}}$ (theorem 5.2.7). We can easily check that the step by step translation at the end of the proof is still possible.

‘$\Leftarrow$’:

As for $\mathsf{K}^{\mathcal{G},2}$ and $\mathsf{K}^{\mathcal{S}}$ (cp. theorem 5.2.7) we can translate the proof step by step (beginning at the root).

5.6.6 THEOREM proofs in $\mathsf{S4}^{\mathcal{H}}$ from proofs in $\mathsf{S4}^{\mathcal{S}}$

$$\mathsf{S4}^{\mathcal{S}} \vdash \epsilon \mid \mathrm{nnf}(A) \quad \Rightarrow \quad \mathsf{S4}^{\mathcal{H}} \vdash A$$

$$
\begin{array}{c}
\vdots \\ \hline
C \lor \Diamond A_1 \lor \ldots \lor \Diamond A_m \\ \hline
\vdots \quad (\mathrm{cpc}), (\mathrm{mp}), (\Diamond\Box_2) \\ \hline
\dfrac{\Box\neg A_1 \to \ldots \to \Box\neg A_m \to C}{\Box(\Box\neg A_1 \to \ldots \to \Box\neg A_m \to C)}\ (\Box) \\ \hline
\vdots \quad (\mathrm{k}), (\mathrm{cpc}), (\mathrm{mp}) \\ \hline
\Box\Box\neg A_1 \to \ldots \to \Box\Box\neg A_m \to \Box C \\ \hline
\vdots \quad (4), (\mathrm{cpc}), (\mathrm{mp}) \\ \hline
\Box\neg A_1 \to \ldots \to \Box\neg A_m \to \Box C \\ \hline
\vdots \quad (\mathrm{cpc}), (\mathrm{mp}), (\Diamond\Box_2) \\ \hline
\Diamond A_1 \lor \ldots \lor \Diamond A_m \lor \Box C \lor B_1 \lor \ldots \lor B_n
\end{array}
$$

Figure 5.m: Simulating the $(\Diamond)$ rule of $\mathsf{S4}^{\mathcal{S}}$ in $\mathsf{S4}^{\mathcal{H}}$.

Proof

We can reuse most parts of the corresponding proof for $\mathsf{KT}^{\mathcal{H}}$ and $\mathsf{KT}^{\mathcal{S}}$. In addition we have to prove that $\mathsf{S4}^{\mathcal{H}} \vdash C \lor \Diamond A_1 \lor \ldots \lor \Diamond A_m$ implies $\mathsf{S4}^{\mathcal{H}} \vdash \Diamond A_1 \lor \ldots \lor \Diamond A_m \lor \Box C \lor B_1 \lor \ldots \lor B_n$. This follows from the proof fragment in figure 5.m.

5.6.7 REMARK from $\mathsf{S4}^{\mathcal{S}}$ to $\mathsf{S4}^{\mathcal{S},2}$

As in the case of $\mathsf{K}^{\mathcal{S}}$ and $\mathsf{KT}^{\mathcal{S}}$, we split the rule $(\Diamond)$ into two rules $(\Diamond, \text{new})$ and $(\Diamond, \text{dup})$. In addition, we no longer set the left hand side to ϵ when we apply $(\Box)$ backwards, but leave it unchanged. This allows us to use Σ instead of $\Diamond\Sigma$ on the right hand side of the premise of the $(\Box)$ rule.

5.6.8 DEFINITION sequent calculus $\mathsf{S4}^{\mathcal{S},2}$

axioms:

$$\frac{}{\Diamond\Sigma \mid \mathsf{true}, \Gamma}\ (\mathsf{true}) \qquad\qquad \frac{}{\Diamond\Sigma \mid P, \neg P, \Gamma}\ (\mathrm{id})$$

rules:

$$\frac{\Diamond\Sigma \mid A, B, \Gamma}{\Diamond\Sigma \mid A \lor B, \Gamma}\ (\lor) \qquad\qquad \frac{\Diamond\Sigma \mid A, \Gamma \quad \Diamond\Sigma \mid B, \Gamma}{\Diamond\Sigma \mid A \land B, \Gamma}\ (\land)$$

$$\frac{\Diamond A, \Diamond\Sigma \mid A, \Gamma}{\Diamond\Sigma \mid \Diamond A, \Gamma}\ \Diamond A \notin \Diamond\Sigma \quad (\Diamond, \text{new}) \qquad\qquad \frac{\Diamond\Sigma \mid A, \Gamma}{\Diamond\Sigma \mid \Diamond A, \Gamma}\ \Diamond A \in \Diamond\Sigma \quad (\Diamond, \text{dup})$$

$$\frac{\Diamond\Sigma \mid A, \Sigma}{\Diamond\Sigma \mid \Box A, \Gamma}\ (\Box)$$

main formulas: true in (true), P and $\neg P$ in (id), $A \lor B$ in $(\lor)$, $A \land B$ in $(\land)$, $\Diamond A$ in $(\Diamond)$, $\Box A$ in $(\Box)$

side formulas: A and B in $(\vee)$, A and B in $(\wedge)$, $\Diamond A$ and A in $(\Diamond, \text{new})$, A in $(\Diamond, \text{dup})$, A in $(\Box)$

5.6.9 THEOREM $\mathsf{S4}^{\mathcal{S},2}$: weakening

If $\mathsf{S4}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$, then $\mathsf{S4}^{\mathcal{S},2} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$. Moreover, if d is the depth of the proof of $\Diamond\Sigma \mid \Gamma$, then there exists a proof of $\Diamond\Delta, \Diamond\Sigma \mid \Pi, \Gamma$ whose depth is at most d.

Proof

The proof is analogous to the corresponding proof for $\mathsf{KT}^{\mathcal{S}}$ (theorem 5.4.3). We only show the case with $(\Box)$.

1. The last step is $\frac{\Diamond\Sigma|A,\Sigma}{\Diamond\Sigma|\Box A,\Gamma}(\Box)$: With the induction hypothesis we obtain $\mathsf{S4}^{\mathcal{S}} \vdash \Diamond\Delta, \Diamond\Sigma \mid A, \Delta, \Sigma$, and with an application of $(\Box)$ follows $\mathsf{S4}^{\mathcal{S}} \vdash \Diamond\Delta, \Diamond\Sigma \mid \Pi, \Box A, \Gamma$.

5.6.10 THEOREM $\mathsf{S4}^{\mathcal{S},2}$: invertible rules

The rule $(\Box)$ of $\mathsf{S4}^{\mathcal{S},2}$ is not invertible. All the other rules are strongly invertible.

Proof

Analogous to the corresponding proof for $\mathsf{K}^{\mathcal{S},2}$ (theorem 5.4.13). We only show $(\Box)\cdots \rightsquigarrow (\Diamond, \text{new})(\Box)\cdots$ and $(\Box)\cdots \rightsquigarrow (\Diamond, \text{dup})(\Box)\cdots$. There we use the first transformation below plus weakening and the second transformation, respectively.

$$\dfrac{\Diamond\Sigma \mid A, \Sigma}{\Diamond\Sigma \mid \Box A, \Diamond B, \Gamma}(\Box) \quad \rightsquigarrow \quad \dfrac{\dfrac{\Diamond B, \Diamond\Sigma \mid A, B, \Sigma}{\Diamond B, \Diamond\Sigma \mid \Box A, B, \Gamma}(\Box)}{\Diamond\Sigma \mid \Box A, \Diamond B, \Gamma}(\Diamond, \text{new})$$

$$\dfrac{\dfrac{\vdots}{\Diamond\Sigma \mid A, \Sigma}}{\Diamond\Sigma \mid \Box A, \Diamond B, \Gamma}(\Box) \quad \rightsquigarrow \quad \dfrac{\dfrac{\dfrac{\vdots}{\Diamond\Sigma \mid A, B, \Sigma}}{\Diamond\Sigma \mid \Box A, B, \Gamma}(\Box)}{\Diamond\Sigma \mid \Box A, \Diamond B, \Gamma}(\Diamond, \text{dup})$$

5.6.11 THEOREM $\mathsf{S4}^{\mathcal{S},2}$: duplicate formulas

$$\mathsf{S4}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, A, \Gamma \quad \Rightarrow \quad \mathsf{S4}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, \Gamma$$

Proof

The proof for $\mathsf{S4}^{\mathcal{S},2}$ is analogous to the corresponding proof for $\mathsf{KT}^{\mathcal{S},2}$ (theorem 5.4.14). We show only the cases where the last step is a $(\Box)$ application.

1. The last step is $\frac{\Diamond\Sigma|B,\Sigma}{\Diamond\Sigma|A,A,\Box B,\Gamma}(\Box)$: With an application of $(\Box)$ we obtain $\mathsf{S4}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, \Box B, \Gamma$.
2. The last step is $\frac{\Diamond\Sigma|B,\Sigma}{\Diamond\Sigma|\Box B,\Box B,\Gamma}(\Box)$, where $A \equiv \Box B$: With an application of $(\Box)$ we obtain $\mathsf{S4}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Box B, \Gamma$.

5.6.12 THEOREM equivalence of $\mathsf{S4}^{\mathcal{S}}$ and $\mathsf{S4}^{\mathcal{S},2}$

$$\mathsf{S4}^{\mathcal{S}} \vdash \epsilon \mid A \quad \Leftrightarrow \quad \mathsf{S4}^{\mathcal{S},2} \vdash \epsilon \mid A$$

Proof

Let $\mathsf{S4}^{\mathcal{S},2-}$ be the calculus $\mathsf{S4}^{\mathcal{S},2}$ with the $(\Box)$ rule replaced by $\frac{\epsilon \mid A, \Diamond\Sigma}{\Diamond\Sigma \mid \Box A, \Gamma}$. The difference between $\mathsf{S4}^{\mathcal{S}}$ and $\mathsf{S4}^{\mathcal{S},2-}$ is the same as the difference between $\mathsf{K}^{\mathcal{S}}$ and $\mathsf{K}^{\mathcal{S},2}$. We can thus proceed as in the proof of theorem 5.2.14 in order to show $\mathsf{S4}^{\mathcal{S}} \vdash \epsilon \mid A \Leftrightarrow \mathsf{S4}^{\mathcal{S},2-} \vdash \epsilon \mid A$.

It remains to show $\mathsf{S4}^{\mathcal{S},2-} \vdash \epsilon \mid A \Leftrightarrow \mathsf{S4}^{\mathcal{S},2} \vdash \epsilon \mid A$.

If $\mathcal{P}$ is a proof in $\mathsf{S4}^{\mathcal{S},2}$, then we use the transformation

$$\begin{array}{c}\vdots\\ \dfrac{\dfrac{\vdots}{\Diamond B_1,\ldots,\Diamond B_n \mid A, B_1,\ldots,B_n}}{\Diamond B_1,\ldots,\Diamond B_n \mid \Box A,\Gamma}(\Box)\\ \vdots\end{array} \quad\rightsquigarrow\quad \begin{array}{c}\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\Diamond B_1,\ldots,\Diamond B_n \mid A, B_1,\ldots,B_n}}{\vdots}(\Diamond,\text{new})}{\Diamond B_1 \mid A, B_1, \Diamond B_2,\ldots,\Diamond B_n}(\Diamond,\text{new})}{\epsilon \mid A, \Diamond B_1,\ldots,\Diamond B_n}(\Diamond,\text{new})}{\Diamond B_1,\ldots,\Diamond B_n \mid \Box A,\Gamma}(\Box)\\ \vdots\end{array}$$

to convert it into a proof of the same sequent in $\mathsf{S4}^{\mathcal{S},2-}$. Thus $\mathsf{S4}^{\mathcal{S},2} \vdash \epsilon \mid A \Rightarrow \mathsf{S4}^{\mathcal{S},2-} \vdash \epsilon \mid A$.

Finally let $\mathcal{P}$ be a proof of the sequent $\Diamond\Sigma \mid \Gamma$ in $\mathsf{S4}^{\mathcal{S},2-}$. We can easily show that the rule $(\Diamond, \text{new})$ is strongly invertible. Thus there exists a proof $\mathcal{P}'$ of $\Diamond\Sigma \mid \Gamma$ in $\mathsf{S4}^{\mathcal{S},2-}$ with the following property: If $\frac{\epsilon \mid A, \Diamond B_1,\ldots,\Diamond B_n}{\Diamond B_1,\ldots,\Diamond B_n \mid \Box A,\Gamma}(\Box)$ is a step in $\mathcal{P}'$, then this $(\Box)$ application is preceded by n applications of $(\Diamond, \text{new})$ with the formulas $B_1,\ldots,B_n$ as main formulas. Using the inverse of the transformation above we can then translate $\mathcal{P}'$ into a proof in $\mathsf{S4}^{\mathcal{S},2}$. Thus $\mathsf{S4}^{\mathcal{S},2-} \vdash \epsilon \mid A \Rightarrow \mathsf{S4}^{\mathcal{S},2} \vdash \epsilon \mid A$.

5.6.13 REMARK $\mathsf{S4}^{\mathcal{S},2}$: non-termination

In general backward proof search in $\mathsf{S4}^{\mathcal{S},2}$ does not terminate.

Example

The simplest example is:

$$\dfrac{\dfrac{\dfrac{\dfrac{\vdots}{\Diamond\Box p_0 \mid p_0, \Box p_0}(\Box)}{\Diamond\Box p_0 \mid p_0, \Box p_0}(\Box)}{\Diamond\Box p_0 \mid \Box p_0}(\Box)}{\epsilon \mid \Diamond\Box p_0}(\Diamond,\text{new})$$

As a second example, we check whether $\text{nnf}(\text{Grz}) \equiv \text{nnf}(\Box(\Box(p_0 \to \Box p_0) \to p_0) \to p_0) \equiv \Diamond(\Box(\neg p_0 \vee \Box p_0) \wedge \neg p_0) \vee p_0$ is provable in $\mathsf{S4}^{\mathcal{S},2}$. We use the abbreviation $A \equiv \Box(\neg p_0 \vee \Box p_0) \wedge \neg p_0$. The proof search does not terminate if we always do the leftmost branch first (see figure 5.n).

5.6.14 REMARK from $\mathsf{S4}^{\mathcal{S},2}$ to $\mathsf{S4}^{\mathcal{S},3}$

Before defining the improved calculus $\mathsf{S4}^{\mathcal{S},3}$, we explain the motivation behind it in terms of backward proof search in $\mathsf{S4}^{\mathcal{S},2}$.

A loop during backward proof search in $\mathsf{S4}^{\mathcal{S},2}$ must contain a $(\Box)$ application, since all the other rules remove connectives. Therefore, it is sufficient to save the sequents we obtain after applying $(\Box)$ backwards with main formula $\Box A$. The history thus has the form $[\Box A_1, \Diamond\Sigma_1]$, $[\Box A_2, \Diamond\Sigma_2]$, …, $[\Box A_m, \Diamond\Sigma_m]$, where $\forall i,j \in \{1,\ldots,m\} : (i \neq j \Rightarrow [\Box A_i, \Diamond\Sigma_i] \neq [\Box A_j, \Diamond\Sigma_j])$ (cp. [Lad77]) and where $\Diamond\Sigma_i$ contains no duplicate formulas. As an example we take the following proof in $\mathsf{S4}^{\mathcal{S},2}$

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{
          \dfrac{
            \dfrac{\vdots \quad \vdots}{\Diamond A \mid p_0, A}\,(\wedge)
          }{\Diamond A \mid \neg p_0, \Box p_0, \Box(\neg p_0 \vee \Box p_0)}\,(\Box)
          \qquad \vdots
        }{\Diamond A \mid \neg p_0, \Box p_0, A}\,(\wedge)
      }{\Diamond A \mid \neg p_0 \vee \Box p_0, A}\,(\vee)
    }{\Diamond A \mid \Box(\neg p_0 \vee \Box p_0), p_0}\,(\Box)
    \qquad
    \dfrac{}{\Diamond A \mid \neg p_0, p_0}\,(\mathrm{id})
  }{\Diamond A \mid A, p_0}\,(\wedge)
}{
  \dfrac{\epsilon \mid \Diamond A, p_0}{\epsilon \mid \Diamond A \vee p_0}\,(\vee)
}\,(\Diamond, \mathrm{new})
$$

Figure 5.n: The proof of the example in remark 5.6.13.

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{
          \dfrac{}{\Diamond p_1, \Diamond A, \Diamond B \mid \neg p_1, p_1, A, B}\,(\mathrm{id})
        }{\Diamond p_1, \Diamond A, \Diamond B \mid \Box\neg p_1, p_1, A, B}\,(\Box)
      }{\Diamond p_1, \Diamond A, \Diamond B \mid p_1, A, B}\,(\Box)
    }{\Diamond A, \Diamond B \mid \Diamond p_1, A, B}\,(\Diamond, \mathrm{new})
  }{\Diamond A, \Diamond B \mid A, B}\,(\Box)
}{\epsilon \mid \Diamond A, \Diamond B}\,(\Diamond, \mathrm{new}),\ (\Diamond, \mathrm{new})
$$

where $A \equiv \Box\Diamond p_1$ and $B \equiv \Box\Box\neg p_1$. So far we would obtain the following proof if we added histories as for example in the case of $(\mathsf{K} + T)^{\mathcal{S},2}$:

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{
          \dfrac{}{[\Box\neg p_1, \Diamond p_1, \Diamond A, \Diamond B], [B, \Diamond p_1, \Diamond A, \Diamond B], [A, \Diamond A, \Diamond B] \mid \Diamond p_1, \Diamond A, \Diamond B \mid \neg p_1, p_1, A, B}\,(\mathrm{id})
        }{[B, \Diamond p_1, \Diamond A, \Diamond B], [A, \Diamond A, \Diamond B] \mid \Diamond p_1, \Diamond A, \Diamond B \mid \Box\neg p_1, p_1, A, B}\,(\Box)
      }{[A, \Diamond A, \Diamond B] \mid \Diamond p_1, \Diamond A, \Diamond B \mid p_1, A, B}\,(\Box)
    }{[A, \Diamond A, \Diamond B] \mid \Diamond A, \Diamond B \mid \Diamond p_1, A, B}\,(\Diamond, \mathrm{new})
  }{\epsilon \mid \Diamond A, \Diamond B \mid A, B}\,(\Box)
}{\epsilon \mid \epsilon \mid \Diamond A, \Diamond B}\,(\Diamond, \mathrm{new}),\ (\Diamond, \mathrm{new})
$$

A $\Diamond$-formula cannot disappear during the backward proof search, i.e. in the history $[\Box A_1, \Diamond\Sigma_1]$, $[\Box A_2, \Diamond\Sigma_2], \ldots, [\Box A_n, \Diamond\Sigma_n]$, we have $\Diamond\Sigma_1 \supseteq \Diamond\Sigma_2 \supseteq \ldots \supseteq \Diamond\Sigma_n$. Therefore we can empty the history if a new $\Diamond$ formula appears, since $\Diamond\Sigma_i \neq \Diamond\Sigma_j$ implies $[\Box A_i, \Diamond\Sigma_i] \neq [\Box A_j, \Diamond\Sigma_j]$. The improved history thus has the form $[\Box B_1, \Diamond\Sigma], [\Box B_2, \Diamond\Sigma], \ldots, [\Box B_n, \Diamond\Sigma]$, where $\forall i, j \in \{1, \ldots, n-1\} : (i \neq j \Rightarrow \Box B_i \neq \Box B_j)$. In the example we obtain:

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{
          \dfrac{}{[\Box\neg p_1, \Diamond p_1, \Diamond A, \Diamond B], [B, \Diamond p_1, \Diamond A, \Diamond B] \mid \Diamond p_1, \Diamond A, \Diamond B \mid \neg p_1, p_1, A, B}\,(\mathrm{id})
        }{[B, \Diamond p_1, \Diamond A, \Diamond B] \mid \Diamond p_1, \Diamond A, \Diamond B \mid \Box\neg p_1, p_1, A, B}\,(\Box)
      }{[A, \Diamond A, \Diamond B] \mid \Diamond p_1, \Diamond A, \Diamond B \mid p_1, A, B}\,(\Box)
    }{[A, \Diamond A, \Diamond B] \mid \Diamond A, \Diamond B \mid \Diamond p_1, A, B}\,(\Diamond, \mathrm{new})
  }{\epsilon \mid \Diamond A, \Diamond B \mid A, B}\,(\Box)
}{\epsilon \mid \epsilon \mid \Diamond A, \Diamond B}\,(\Diamond, \mathrm{new}),\ (\Diamond, \mathrm{new})
$$

But if every element of the history contains the same $\Diamond$ formulas, they have no effect and thus we can remove them. The further improved history is thus a simple list $\Box C_1, \ldots, \Box C_n$, where $\forall i, j \in \{1, \ldots, n\} : (i \neq j \Rightarrow \Box C_i \neq \Box C_j)$. The example simplifies to

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Box\neg p_1, B \mid \Diamond p_1, \Diamond A, \Diamond B \mid \neg p_1, p_1, A, B}\,(\mathrm{id})}{B \mid \Diamond p_1, \Diamond A, \Diamond B \mid \Box\neg p_1, p_1, A, B}\,(\Box)}{A \mid \Diamond p_1, \Diamond A, \Diamond B \mid p_1, A, B}\,(\Box)}{A \mid \Diamond A, \Diamond B \mid \Diamond p_1, A, B}\,(\Diamond, \mathrm{new})}{\epsilon \mid \Diamond A, \Diamond B \mid A, B}\,(\Box)}{\epsilon \mid \epsilon \mid \Diamond A, \Diamond B}\,(\Diamond, \mathrm{new}), (\Diamond, \mathrm{new})$$

See remark 5.6.18 for a formula where it is essential that we empty the history when we find a new $\Diamond$ formula.

So far we have motivated the calculus $\mathsf{S4}^{\mathcal{S},3}$ without the multiset $\Box\Pi$ in the premise of $(\Box)$ (cp. definition 5.6.15). Consider for example the following proof in $\mathsf{S4}^{\mathcal{S},2}$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Diamond\Box p_1, \Diamond\Box\mathsf{true} \mid \mathsf{true}, \Box p_1, \Box\mathsf{true}}\,(\mathsf{true})}{\Diamond\Box p_1, \Diamond\Box\mathsf{true} \mid p_1, \Box p_1, \Box\mathsf{true}}\,(\Box)}{\Diamond\Box p_1, \Diamond\Box\mathsf{true} \mid \Box p_1, \Box\mathsf{true}}\,(\Box)}{\Diamond\Box p_1 \mid \Box p_1, \Diamond\Box\mathsf{true}}\,(\Diamond, \mathrm{new})}{\epsilon \mid \Diamond\Box p_1, \Diamond\Box\mathsf{true}}\,(\Diamond, \mathrm{new})$$

If we add the histories as described so far we obtain

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Box\mathsf{true}, \Box p_1 \mid \Diamond\Box p_1, \Diamond\Box\mathsf{true} \mid \mathsf{true}, \Box p_1, \Box\mathsf{true}}\,(\mathsf{true})}{\Box p_1 \mid \Diamond\Box p_1, \Diamond\Box\mathsf{true} \mid p_1, \Box p_1, \Box\mathsf{true}}\,(\Box)}{\epsilon \mid \Diamond\Box p_1, \Diamond\Box\mathsf{true} \mid \Box p_1, \Box\mathsf{true}}\,(\Box)}{\epsilon \mid \Diamond\Box p_1 \mid \Box p_1, \Diamond\Box\mathsf{true}}\,(\Diamond, \mathrm{new})}{\epsilon \mid \epsilon \mid \Diamond\Box p_1, \Diamond\Box\mathsf{true}}\,(\Diamond, \mathrm{new})$$

This proof is in fact too complicated, as the application of $(\Box)$ with main formula $\Box p_1$ is superfluous. From the point of view of backward proof search, we could obtain the sequent with the multiset $\mathsf{true}, \Box p_1, \Box\mathsf{true}$ if we chose $\Box\mathsf{true}$ as the main formula in the first backward application of $(\Box)$. Then we would find the proof

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Box\mathsf{true} \mid \Diamond\Box p_1, \Diamond\Box\mathsf{true} \mid \mathsf{true}, \Box p_1, \Box\mathsf{true}}\,(\mathsf{true})}{\epsilon \mid \Diamond\Box p_1, \Diamond\Box\mathsf{true} \mid p_1, \Box p_1, \Box\mathsf{true}}\,(\Box)}{\epsilon \mid \Diamond\Box p_1 \mid \Box p_1, \Diamond\Box\mathsf{true}}\,(\Diamond, \mathrm{new})}{\epsilon \mid \epsilon \mid \Diamond\Box p_1, \Diamond\Box\mathsf{true}}\,(\Diamond, \mathrm{new})$$

This is a special case of a situation we encounter frequently during backward proof search. Assume that we apply $(\Box)$ backwards on the sequent $\Box H \mid \Diamond\Sigma \mid \Box A, \Box B, \Gamma$ and obtain the new sequent $\Box A, \Box H \mid \Diamond\Sigma \mid A, \Sigma$. Assume further that after several backwards applications of rules we obtain the sequent $\Box H', \Box A, \Box H \mid \Diamond\Sigma \mid \Box B, \Gamma'$. Then we do not have to apply $(\Box)$ backwards with B as the main formula, because the resulting sequent $\Box B, \Box H', \Box A, \Box H \mid \Diamond\Sigma \mid B, \Sigma$ (with a shorter history) can be obtained directly from the original sequent $\Box H \mid \Diamond\Sigma \mid \Box A, \Box B, \Gamma$. Therefore we can put $\Box B$ into the history when we apply $(\Box)$ with main formula $\Box A$ backwards on the original sequence. This is the reason for the multiset $\Box\Pi$ in the $(\Box)$ rule of $\mathsf{S4}^{\mathcal{S},3}$. See also theorem 6.5.5 for the effects of $\Box\Pi$.

5.6.15 DEFINITION sequent calculus $\mathsf{S4}^{\mathcal{S},3}$

axioms:

$$\frac{}{\Box H \mid \Diamond\Sigma \mid \mathsf{true}, \Gamma}\ (\mathsf{true}) \qquad \frac{}{\Box H \mid \Diamond\Sigma \mid P, \neg P, \Gamma}\ (\mathrm{id})$$

rules:

$$\frac{\Box H \mid \Diamond\Sigma \mid A, B, \Gamma}{\Box H \mid \Diamond\Sigma \mid A \vee B, \Gamma}\ (\vee) \qquad \frac{\Box H \mid \Diamond\Sigma \mid A, \Gamma \quad \Box H \mid \Diamond\Sigma \mid B, \Gamma}{\Box H \mid \Diamond\Sigma \mid A \wedge B, \Gamma}\ (\wedge)$$

$$\frac{\epsilon \mid \Diamond A, \Diamond\Sigma \mid A, \Gamma}{\Box H \mid \Diamond\Sigma \mid \Diamond A, \Gamma}\ \Diamond A \notin \Diamond\Sigma \quad (\Diamond, \mathrm{new}) \qquad \frac{\Box H \mid \Diamond\Sigma \mid A, \Gamma}{\Box H \mid \Diamond\Sigma \mid \Diamond A, \Gamma}\ \Diamond A \in \Diamond\Sigma \quad (\Diamond, \mathrm{dup})$$

$$\frac{\Box A, \Box\Pi, \Box H \mid \Diamond\Sigma \mid A, \Sigma}{\Box H \mid \Diamond\Sigma \mid \Box A, \Gamma}\ \begin{array}{l} \Box A \notin \Box H \\ \Box C \in \Gamma \Rightarrow \Box C \in \Box A, \Box\Pi, \Box H \\ \text{no duplicate elements in } \Box A, \Box\Pi, \Box H \end{array} \quad (\Box)$$

main formulas: true in (true), P and $\neg P$ in (id), $A \vee B$ in $(\vee)$, $A \wedge B$ in $(\wedge)$, $\Diamond A$ in $(\Diamond, \mathrm{new})$ and $(\Diamond, \mathrm{dup})$, $\Box A$ in $(\Box)$

side formulas: A and B in $(\vee)$, A and B in $(\wedge)$, A and $\Diamond A$ in $(\Diamond, \mathrm{new})$, A in $(\Diamond, \mathrm{dup})$, A in $(\Box)$

5.6.16 THEOREM equivalence of $\mathsf{S4}^{\mathcal{S},2}$ and $\mathsf{S4}^{\mathcal{S},3}$

$$\mathsf{S4}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma \quad \Leftrightarrow \quad \mathsf{S4}^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma$$

Proof

'$\Leftarrow$':

Let $\mathcal{P}$ be a proof of $\epsilon \mid \Diamond\Sigma \mid \Gamma$ in $\mathsf{S4}^{\mathcal{S},3}$. We remove all histories from $\mathcal{P}$. With an induction on proof depth we can easily show that the result is a proof of $\Diamond\Sigma \mid \Gamma$ in $\mathsf{S4}^{\mathcal{S},2}$.

'$\Rightarrow$':

The auxiliary calculus $\mathsf{S4}^{\mathcal{S},2+}$ is the calculus $\mathsf{S4}^{\mathcal{S},2}$ with the $(\Box)$ rule replaced by the rule

$$\frac{\Box A, \Box H \mid \Diamond\Sigma \mid A, \Sigma}{\Box H \mid \Diamond\Sigma \mid \Box A, \Gamma}\ \Box A \notin \Box H \quad (\Box)$$

We prove the following:

$$\mathsf{S4}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma \ \Rightarrow\ \mathsf{S4}^{\mathcal{S},2+} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma \ \Rightarrow\ \mathsf{S4}^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma$$

1. $\mathsf{S4}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma \;\Rightarrow\; \mathsf{S4}^{\mathcal{S},2+} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma$:

This step is analogous to the step from $(\mathsf{K}+T)^{\mathcal{S},2}$ to $(\mathsf{K}+T)^{\mathcal{S},3}$: We first remove the duplicate occurrences of sequents on a branch and then translate the proof step by step.

Let $\mathcal{P}$ be a proof of $\Diamond\Sigma \mid \Gamma$ in $\mathsf{S4}^{\mathcal{S},2}$. If a sequent S occurs twice on a branch in $\mathcal{P}$, then we remove the part of the branch between the two occurrences together with one occurrence of S. We repeat this transformation as long as possible. Let $\mathcal{P}'$ be the resulting proof of $\Diamond\Sigma \mid \Gamma$ in $\mathsf{S4}^{\mathcal{S},2}$. Beginning at the root of $\mathcal{P}'$ we add histories to the sequents (cp. the proof of theorem 5.3.15). The result is a proof of $\Diamond\Sigma \mid \Gamma$ in $\mathsf{S4}^{\mathcal{S},2+}$.

2. $\mathsf{S4}^{\mathcal{S},2+} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma \;\Rightarrow\; \mathsf{S4}^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma$:

Let $\mathcal{P}$ be a proof of $\epsilon \mid \Diamond\Sigma \mid \Gamma$ in $\mathsf{S4}^{\mathcal{S},2+}$. We use the motivation from remark 5.6.14. If $\mathcal{P}$ is not already a proof in $\mathsf{S4}^{\mathcal{S},3}$, then the following structure must occur in $\mathcal{P}$:

$$\left.\begin{array}{c}
\dfrac{\Box B, \Box A, \Box H', \Box H \mid \Diamond\Sigma \mid B, \Sigma}{\Box A, \Box H', \Box H \mid \Diamond\Sigma \mid \Box A, \Gamma'}\,(\Box) \\
\hline
\vdots \\
\hline
\dfrac{\Box A, \Box H \mid \Diamond\Sigma \mid A, \Sigma}{\Box H \mid \Diamond\Sigma \mid \Box A, \Box B, \Gamma}\,(\Box)
\end{array}\right\} (*)$$

where $\Box A \notin \Box H$, $\Box B \notin \Box H$, and no $(\Diamond, \text{new})$ application occurs in $(*)$. We replace this structure in $\mathcal{P}$ by

$$\frac{\Box B, \Box H \mid \Diamond\Sigma \mid B, \Sigma}{\Box H \mid \Diamond\Sigma \mid \Box A, \Box B, \Gamma}\,(\Box)$$

and adapt the histories above the sequent $\Box B, \Box H \mid \Diamond\Sigma \mid B, \Sigma$. The result is still a proof in $\mathsf{S4}^{\mathcal{S},2+}$.

In this way we eliminate all the $(\Box)$ applications in $\mathcal{P}$ that are not allowed as $(\Box)$ applications in $\mathsf{S4}^{\mathcal{S},3}$. The result is a proof of $\epsilon \mid \Diamond\Sigma \mid \Gamma$ in $\mathsf{S4}^{\mathcal{S},2+}$ as well as in $\mathsf{S4}^{\mathcal{S},3}$.

5.6.17 THEOREM $\mathsf{S4}^{\mathcal{S},3}$: termination

Backward proof search in $\mathsf{S4}^{\mathcal{S},3}$ for sequents of the form $\epsilon \mid \epsilon \mid A$ always terminates.

Proof

We assume that we do backward proof search for the sequent $\epsilon \mid \epsilon \mid A$.

We define $m(\Box H \mid \Diamond\Sigma \mid \Gamma) := c^3 \cdot (c - \text{card}(\Diamond\Sigma)) + c^2 \cdot (c - \text{card}(\Box H)) + \text{length}(\Gamma)$, where $c := \text{length}(A)$.

For every sequent $\Box H \mid \Diamond\Sigma \mid \Gamma$ that can occur during the proof search for $\epsilon \mid \epsilon \mid A$, we have $m(\Box H \mid \Diamond\Sigma \mid \Gamma) \geq 0$.

The value computed by m decreases with each backward application of one of the rules of $\mathsf{S4}^{\mathcal{S},2}$:

1. $m(\Box H \mid \Diamond\Sigma \mid A \vee B, \Gamma) > m(\Box H \mid \Diamond\Sigma \mid A, B, \Gamma)$
2. $m(\Box H \mid \Diamond\Sigma \mid A \wedge B, \Gamma) > m(\Box H \mid \Diamond\Sigma \mid A, \Gamma)$,
   $m(\Box H \mid \Diamond\Sigma \mid A \wedge B, \Gamma) > m(\Box H \mid \Diamond\Sigma \mid B, \Gamma)$
3. $m(\Box H \mid \Diamond\Sigma \mid \Diamond A, \Gamma) > m(\Box H \mid \Diamond\Sigma \mid \Gamma)$
4. $m(\Box H \mid \Diamond\Sigma \mid \Diamond A, \Gamma) - m(\epsilon \mid \Diamond A, \Diamond\Sigma \mid A, \Gamma)$
   $= c^3 \cdot 1 - c^2 \cdot \text{card}(\Box H) + 1$
   $\geq c^3 - c^3 + 1$
   $= 1$

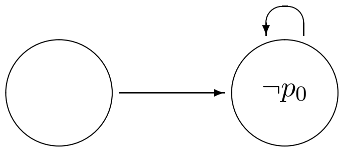
5. $m(\Box H \mid \Diamond\Sigma \mid \Box A, \Gamma) - m(\Box A, \Box\Pi, \Box H \mid \Diamond\Sigma \mid A, \Sigma)$
   $= c^3 \cdot 0 + c^2 \cdot (1 + \mathrm{card}(\Box\Pi)) + (1 + \mathrm{length}(\Gamma) - \mathrm{length}(\Diamond\Sigma))$
   $\geq c^2 + 1 - c^2$
   ≥ 1

Example

In $\mathsf{S4}^{\mathcal{S},3}$, backward proof search for $\Diamond\Box p_0$ fails without looping, in contrast to the example in remark 5.6.13, since no rule is applicable on the sequent $\Box p_0 \mid \Diamond\Box p_0 \mid p_0, \Box p_0$.

$$\dfrac{\dfrac{\Box p_0 \mid \Diamond\Box p_0 \mid p_0, \Box p_0}{\epsilon \mid \Diamond\Box p_0 \mid \Box p_0}\,(\Box)}{\epsilon \mid \epsilon \mid \Diamond\Box p_0}\,(\Diamond, \text{new})$$

We can extract the following countermodel:



5.6.18 REMARK $(\Diamond, \text{new})$

The backward proof search for the formula $\Diamond\Box(p_0 \vee \Box\Diamond\neg p_0)$ shows that we cannot replace ϵ by H in the premise of the $(\Diamond, \text{new})$ rule, i.e. we have to remove the history in such situations. Note that there are no backtracking points in this search. We use the abbreviation $A \equiv \Box(p_0 \vee \Box\Diamond\neg p_0)$.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{A \mid \Diamond\neg p_0, \Diamond A \mid p_0, \Box\Diamond\neg p_0, \neg p_0, A}\,(\mathrm{id})}{A \mid \Diamond\neg p_0, \Diamond A \mid p_0 \vee \Box\Diamond\neg p_0, \neg p_0, A}\,(\vee)}{\epsilon \mid \Diamond\neg p_0, \Diamond A \mid \neg p_0, A}\,(\Box)}{\Box\Diamond\neg p_0, A \mid \Diamond A \mid \Diamond\neg p_0, A}\,(\Diamond, \text{new})}{A \mid \Diamond A \mid p_0, \Box\Diamond\neg p_0, A}\,(\Box)}{A \mid \Diamond A \mid p_0 \vee \Box\Diamond\neg p_0, A}\,(\vee)}{\epsilon \mid \Diamond A \mid A}\,(\Box)}{\epsilon \mid \epsilon \mid \Diamond A}\,(\Diamond, \text{new})$$

5.6.19 THEOREM proofs in $\mathsf{S4}^{\mathcal{S},3}$

If $\mathsf{S4}^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma$, then there exists a proof of $\epsilon \mid \Diamond\Sigma \mid \Gamma$ in $\mathsf{S4}^{\mathcal{S},3}$ such that $(\Box)$ is only applied backwards if no other rule is applicable backwards.

Proof

Because of theorem 5.6.10 and theorem 5.6.16 there exists a proof of $\Diamond\Sigma \mid \Gamma$ in $\mathsf{S4}^{\mathcal{S},2}$ such that $(\Box)$ is only applied backwards if no other rule is applicable backwards. We now use the construction from the proof of theorem 5.6.16 to transform it into a proof of $\epsilon \mid \Diamond\Sigma \mid \Gamma$ in $\mathsf{S4}^{\mathcal{S},3}$ of the desired form.

5.6.20 THEOREM $\mathsf{S4}^{\mathcal{S},3}$: permutation of rules

For the calculus $\mathsf{S4}^{\mathcal{S},3}$, the following applies for all axioms x and all rules r:

- If r is not $(\Box)$, then $x \rightsquigarrow rx$.
- If neither r nor s is $(\Box)$, then $rs\cdots \rightsquigarrow sr\cdots$.

Proof

Since the only difference between $\mathsf{S4}^{\mathcal{S},2}$ and $\mathsf{S4}^{\mathcal{S},3}$ is in the $(\Box)$ rule, we can use the same transformations as in the proof of theorem 5.6.10.

5.6.21 REMARK invertible rules in $\mathsf{S4}^{\mathcal{S},3}$

The two theorems 5.6.19 and 5.6.20 show that if we do backward proof search in $\mathsf{S4}^{\mathcal{S},3}$ for a sequent of the form $\epsilon \mid \epsilon \mid A$, then we can do this as if the rules $(\vee)$, $(\wedge)$, $(\Diamond, \text{new})$, $(\Diamond, \text{dup})$ were invertible: We apply these rules backwards as long as possible, and the order in which we do this does not matter.

5.6.22 THEOREM $\mathsf{S4}^{\mathcal{S},3}$: empty the history

$$\mathsf{S4}^{\mathcal{S},3} \vdash H \mid \Diamond\Sigma \mid \Gamma \quad \Rightarrow \quad \mathsf{S4}^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma$$

Proof

Let $\mathcal{P}$ be the proof $H \mid \Diamond\Sigma \mid \Gamma$ in $\mathsf{S4}^{\mathcal{S},3}$. Beginning at $H \mid \Diamond\Sigma \mid \Gamma$, we remove all elements of H from the left hand side of the sequents in $\mathcal{P}$. On each branch we stop as soon as we reach an empty history. With an induction on proof depth we can easily show that the result is a proof of $\epsilon \mid \Diamond\Sigma \mid \Gamma$ in $\mathsf{S4}^{\mathcal{S},3}$.

5.6.23 THEOREM $\mathsf{S4}^{\mathcal{S},3}$: extend (id)

If A is in negation normal form, then:

$$\mathsf{S4}^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid A, \mathrm{nnf}(\neg A), \Gamma$$

Proof

With an induction on length(A) we prove $\mathsf{S4}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, \mathrm{nnf}(\neg A), \Gamma$. We only show the case $A \equiv \Box B$.

1. $A \equiv \Box B$: If $\Diamond\mathrm{nnf}(\neg B) \in \Diamond\Sigma$, then $\mathsf{S4}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid B, \Sigma$ because of the induction hypothesis, and with a $(\Box)$ application we obtain $\mathsf{S4}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Box B, \Diamond\mathrm{nnf}(\neg B), \Gamma$. If $\Diamond\mathrm{nnf}(\neg B) \notin \Diamond\Sigma$, then $\mathsf{S4}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid B, \mathrm{nnf}(\neg B), \Sigma$ because of the induction hypothesis. With a $(\Box)$ application we obtain $\mathsf{S4}^{\mathcal{S},2} \vdash \Diamond\mathrm{nnf}(\neg B), \Diamond\Sigma \mid \Box B, \Gamma$, and with a $(\Diamond)$ application $\mathsf{S4}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Box B, \Diamond\mathrm{nnf}(\neg B), \Gamma$.

Now we use use theorem 5.6.16 to obtain $\mathsf{S4}^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid A, \mathrm{nnf}(\neg A), \Gamma$. Note that the history must be empty, since for example $\mathsf{S4}^{\mathcal{S},3} \nvdash \Box p_0 \mid \epsilon \mid \Box p_0, \Diamond\neg p_0$.

5.6.24 THEOREM $\mathsf{S4}^{\mathcal{S},3}$: duplicate formulas

$$\mathsf{S4}^{\mathcal{S},3} \vdash H \mid \Diamond\Sigma \mid A, A, \Gamma \quad \Rightarrow \quad \mathsf{S4}^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid A, \Gamma$$

Proof

We proceed as in the corresponding proof for $(\mathsf{K}+T)^{\mathcal{S},3}$ (theorem 5.3.10).

With theorem 5.6.22 we obtain $\mathsf{S4}^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid A, A, \Gamma$ and with theorem 5.6.16 $\mathsf{S4}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, A, \Gamma$. Thus $\mathsf{S4}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid A, \Gamma$ because of theorem 5.6.11. With theorem 5.6.16 we finally obtain $\mathsf{S4}^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid A, \Gamma$.

5.6.25 THEOREM $\mathsf{S4}^{\mathcal{S},3}$: use-check

Assume that we do backward proof search for a sequent of the form $\epsilon \mid \epsilon \mid D$. Let $H \mid \Diamond\Sigma \mid A, \Gamma$ be a sequent we obtain during this search, and let $\mathcal{P}$ be a proof in $\mathsf{S4}^{\mathcal{S},3}$ of this sequent. We mark it as follows:

- The main formulas in the instances of axioms are marked.
- If a side formula is marked, then the main formula is marked.
- If in the premise of an instance of $(\vee)$, $(\wedge)$, $(\Diamond, \text{new})$ or $(\Diamond, \text{dup})$ a formula in Γ is marked, then the corresponding formula in the multiset Γ in the conclusion is marked.
- If in the premise of an instance of $(\vee)$, $(\wedge)$, $(\Diamond, \text{new})$ or $(\Diamond, \text{dup})$ a formula in $\Diamond\Sigma$ is marked, then the corresponding formula in the multiset $\Diamond\Sigma$ in the conclusion is marked.
- If in the premise of an instance of $(\Box)$ a formula in Σ or $\Diamond\Sigma$ is marked, then the corresponding formula in the multiset $\Diamond\Sigma$ in the conclusion is marked.

If the formula A in the sequent $H \mid \Diamond\Sigma \mid A, \Gamma$ is not marked, then $\mathsf{S4}^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma$.

Proof

We prove the theorem for $\mathsf{S4}^{\mathcal{S},2}$, defining marked proofs in $\mathsf{S4}^{\mathcal{S},2}$ as described above. Now we take the marked proof $\mathcal{P}$ of $H \mid \Diamond\Sigma \mid A, \Gamma$ and omit all histories, but do not change the marks. We call the result $\mathcal{P}'$. With an induction on proof depth we can easily show that $\mathcal{P}'$ is a marked proof of $\Diamond\Sigma \mid A, \Gamma$ in $\mathsf{S4}^{\mathcal{S},2}$. Thus A is not marked, and therefore $\mathsf{S4}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$. With theorem 5.3.15 we obtain $\mathsf{S4}^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma$.

5.6.26 REMARK $\mathsf{S4}^{\mathcal{S},3}$: use-check

The preceding theorem is no longer true if we replace $\epsilon \mid \Diamond\Sigma \mid \Gamma$ by $\Box H \mid \Diamond\Sigma \mid \Gamma$.

Example

We do backward proof search in $\mathsf{S4}^{\mathcal{S},3}$ for the sequent $\epsilon \mid \epsilon \mid \Diamond\Box p_0, \Diamond\neg p_0, \Diamond(\Diamond p_1 \wedge \Box p_2)$ using the abbreviation $\Sigma = \Box p_0, \neg p_0, \Diamond p_1 \wedge \Box p_2$. See figure 5.o for the resulting proof. The left branch of the right branch ends in an instance of the (id) axiom. Because of use-check we could thus cut off the right branch of the right branch. However, $\mathsf{S4}^{\mathcal{S},3} \nvdash \Box p_2, \Box p_0 \mid \Diamond\Sigma \mid p_2, \Box p_0, \neg p_0, \Box p_2$ because of the condition in the $(\Box)$ rule. Note that it is nevertheless correct to cut off this branch during proof search.

5.6.27 THEOREM (empty) is admissible

If $\epsilon \mid \Diamond\Sigma \mid \Gamma$ is provable in the calculus $\mathsf{S4}^{\mathcal{S},3}$ plus the rule $\frac{\epsilon \mid \Diamond\Sigma \mid \Gamma}{H \mid \Diamond\Sigma \mid \Gamma}(\text{empty})$, then $\mathsf{S4}^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma$.

Proof

Let $\mathcal{P}$ be a proof of $\epsilon \mid \Diamond\Sigma \mid \Gamma$ in $\mathsf{S4}^{\mathcal{S},3}$ plus the rule (empty). If we remove the applications of (empty) and all histories, then we obtain a proof of $\Diamond\Sigma \mid \Gamma$ in $\mathsf{S4}^{\mathcal{S},2}$, i.e. $\mathsf{S4}^{\mathcal{S},2} \vdash \Diamond\Sigma \mid \Gamma$. With theorem 5.6.16 $\mathsf{S4}^{\mathcal{S},3} \vdash \epsilon \mid \Diamond\Sigma \mid \Gamma$.

$$
\dfrac{
\dfrac{\vdots}{\epsilon \mid \Diamond\Sigma \mid \Box p_0, \neg p_0, \Diamond p_1}\,(\Box)
\qquad
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{}{\Box p_0 \mid \Diamond p_1, \Diamond\Sigma \mid p_0, \Box p_0, \neg p_0, \Diamond p_1 \wedge \Box p_2}\,(\mathrm{id})
}{\epsilon \mid \Diamond p_1, \Diamond\Sigma \mid p_2, \Box p_0, \neg p_0, p_1}\,(\Box)
}{\Box p_2, \Box p_0 \mid \Diamond\Sigma \mid p_2, \Box p_0, \neg p_0, \Diamond p_1}\,(\Diamond, \mathrm{new})
\qquad
\Box p_2, \Box p_0 \mid \Diamond\Sigma \mid p_2, \Box p_0, \neg p_0, \Box p_2
}{\Box p_2, \Box p_0 \mid \Diamond\Sigma \mid p_2, \Box p_0, \neg p_0, \Diamond p_1 \wedge \Box p_2}\,(\wedge)
}{\epsilon \mid \Diamond\Sigma \mid \Box p_0, \neg p_0, \Box p_2}\,(\Box)
}{
\dfrac{
\dfrac{
\dfrac{\epsilon \mid \Diamond\Sigma \mid \Box p_0, \neg p_0, \Diamond p_1 \wedge \Box p_2}{\epsilon \mid \Diamond\Box p_0, \Diamond\neg p_0 \mid \Box p_0, \neg p_0, \Diamond(\Diamond p_1 \wedge \Box p_2)}\,(\Diamond, \mathrm{new})
}{\epsilon \mid \Diamond\Box p_0 \mid \Box p_0, \Diamond\neg p_0, \Diamond(\Diamond p_1 \wedge \Box p_2)}\,(\Diamond, \mathrm{new})
}{\epsilon \mid \epsilon \mid \Diamond\Box p_0, \Diamond\neg p_0, \Diamond(\Diamond p_1 \wedge \Box p_2)}\,(\Diamond, \mathrm{new})
}\,(\wedge)
$$

Figure 5.o: The proof of the example in remark 5.6.26.

5.6.28 REMARK using optimisations during proof search in $\mathsf{S4}^{\mathcal{S},3}$

The preceding theorem has the same purpose: It makes it possible to apply the theorems 5.6.23, 5.6.24, 5.6.25 during backward proof search even if the history is not empty. The only condition is that we started with a sequent of the form $\epsilon \mid \epsilon \mid A$. The idea is the same as in the case of $(\mathsf{K}+T)^{\mathcal{S},3}$; therefore we just refer to remark 5.3.25.

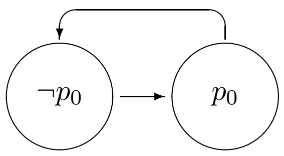
5.6.29 REMARK $\mathsf{S4}^{\mathcal{S},3}$: backward proof search

If a search fails, then we choose a subtree of the search tree that is sufficient to show that the search fails, and extract a countermodel from this subtree. If a branch of this subtree fails because of the condition in the $(\Box)$ rule, then the corresponding branch of the countermodel has a loop.

Example

We do backward proof search in $\mathsf{S4}^{\mathcal{S},3}$ for the sequent $\epsilon \mid \epsilon \mid \Diamond(\Box(\neg p_0 \vee \Box p_0) \wedge \neg p_0) \vee p_0$. In contrast to backward proof search in $\mathsf{S4}^{\mathcal{S},2}$, the search terminates (cp. the second example of remark 5.6.13). We obtain the non-proof in figure 5.p, where $A \equiv \Box(\neg p_0 \vee \Box p_0) \wedge \neg p_0$. Note that there is no possibility for backtracking because of the condition in the $(\Box)$ rule of $\mathsf{S4}^{\mathcal{S},3}$.

The countermodel that can be extracted from the two failing branches is both times



5.7 K_t

5.7.1 REMARK important preliminary remark

The work in this section is to some extent work in progress. We define several calculi and state some theorems, but the proofs are only sketched, since writing them down in detail would make this chapter far too long. (A technical report is planned). We hope that the motivation for the calculi becomes nevertheless clear.

5.7.2 REMARK from $\mathsf{K}_t^{\mathcal{G}}$ to $\mathsf{K}_t^{\mathcal{S}}$

Although $\mathsf{K}_t^{\mathcal{G},2}$ and $\mathsf{K}_t^{\mathcal{G},3}$ are graph calculi, we define them in this chapter since their only purpose is to make the step from $\mathsf{K}_t^{\mathcal{G}}$ to $\mathsf{K}_t^{\mathcal{S}}$ easier to understand.

We prove the equivalence of $\mathsf{K}_t^{\mathcal{G}}$, $\mathsf{K}_t^{\mathcal{G},2}$, $\mathsf{K}_t^{\mathcal{G},3}$, $\mathsf{K}_t^{\mathcal{S}}$. The difficult step is the one from $\mathsf{K}_t^{\mathcal{G},2}$ to $\mathsf{K}_t^{\mathcal{G},3}$. The graph calculus $\mathsf{K}_t^{\mathcal{G},2}$ is essentially a notational variant of the graph calculus $\mathsf{K}_t^{\mathcal{G}}$, and the sequent $\mathsf{K}_t^{\mathcal{S}}$ is obtained in the same way from $\mathsf{K}_t^{\mathcal{G},3}$ as we obtained $\mathsf{K}^{\mathcal{S}}$ from $\mathsf{K}^{\mathcal{G},2}$.

5.7.3 CONVENTION F

F is a formula of K_t or the natural number 0.

5.7.4 REMARK from $\mathsf{K}_t^{\mathcal{G}}$ to $\mathsf{K}_t^{\mathcal{G},2}$

There are three notational differences between $\mathsf{K}_t^{\mathcal{G}}$ and $\mathsf{K}_t^{\mathcal{G},2}$:

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{
          \dfrac{
            \square p_0, \square(\neg p_0 \lor \square p_0) \mid \Diamond A \mid p_0, \square(\neg p_0 \lor \square p_0)
            \quad
            \dfrac{}{\square p_0, \square(\neg p_0 \lor \square p_0) \mid \Diamond A \mid p_0, \neg p_0}\,(\mathrm{id})
          }{\square p_0, \square(\neg p_0 \lor \square p_0) \mid \Diamond A \mid p_0, A}\,(\wedge)
        }{\square(\neg p_0 \lor \square p_0) \mid \Diamond A \mid \neg p_0, \square p_0, \square(\neg p_0 \lor \square p_0)}\,(\square)
        \quad \vdots
      }{\square(\neg p_0 \lor \square p_0) \mid \Diamond A \mid \neg p_0, \square p_0, A}\,(\wedge)
    }{\square(\neg p_0 \lor \square p_0) \mid \Diamond A \mid \neg p_0 \lor \square p_0, A}\,(\vee)
  }{\epsilon \mid \Diamond A \mid \square(\neg p_0 \lor \square p_0), p_0}\,(\square)
  \quad
  \dfrac{}{\epsilon \mid \Diamond A \mid \neg p_0, p_0}\,(\mathrm{id})
}{
  \dfrac{
    \dfrac{\epsilon \mid \Diamond A \mid A, p_0}{\epsilon \mid \epsilon \mid \Diamond A, p_0}\,(\Diamond, \mathrm{new})
  }{\epsilon \mid \epsilon \mid \Diamond A \lor p_0}\,(\vee)
}\,(\wedge)
$$

Figure 5.p: The non-proof of the example in 5.6.29.

- If we do backward proof search in $\mathsf{K}_t^{\mathcal{G},2}$ for a graph of the form $\boxed{\boxed{\epsilon \mid A}}$, then only graphs in the form of trees can occur.

  The root is the leftmost node of the tree. In order to maintain an unambiguous notation each arrow is labelled with the formula that created it. An arrow labelled with a $\Box$ formula thus corresponds to an arrow from left to right in $\mathsf{K}_t^{\mathcal{G}}$, and an arrow labelled with a $\blacksquare$ formula corresponds to an arrow from right to left in $\mathsf{K}_t^{\mathcal{G}}$.
- Each formula A has an index F, the 'origin' of A. Assume that A^F is in a world w. If F is 0 this means that A came from the left hand side, i.e. from the father of w, or from w itself. Otherwise A came from the right hand side, i.e. from a child of w, and F is the label of the arrow from w to this child.
- If a label of an arrow is written in parentheses, for example $[D]$, then there is either no such arrow or an arrow labelled with the formula D.

In addition, we add conditions to the rules such that a rule is applicable only once in a certain world with a certain main formula.

5.7.5 DEFINITION graph calculus $\mathsf{K}_t^{\mathcal{G},2}$

axioms:

$$\frac{}{\cdots \xrightarrow{[D]} \boxed{\boxed{\Delta \mid \mathsf{true}^F, \Gamma}} \xrightarrow{*} \cdots}\ (\mathsf{true})$$

$$\frac{}{\cdots \xrightarrow{[D]} \boxed{\boxed{\Delta \mid P^{F_1}, (\neg P)^{F_2}, \Gamma}} \xrightarrow{*} \cdots}\ (\mathrm{id})$$

rules:

$$\frac{\cdots \xrightarrow{[D]} \boxed{\boxed{(A \vee B)^F, \Delta \mid A^0, B^0, \Gamma}} \xrightarrow{*} \cdots}{\cdots \xrightarrow{[D]} \boxed{\boxed{\Delta \mid (A \vee B)^F, \Gamma}} \xrightarrow{*} \cdots}\ (\vee)$$

where: $\forall F' : (A \vee B)^{F'} \notin \Delta$

$$\frac{\cdots \xrightarrow{[D]} \boxed{\boxed{(A \wedge B)^F, \Delta \mid A^0, \Gamma}} \xrightarrow{*} \cdots \qquad \cdots \xrightarrow{[D]} \boxed{\boxed{(A \wedge B)^F, \Delta \mid B^0, \Gamma}} \xrightarrow{*} \cdots}{\cdots \xrightarrow{[D]} \boxed{\boxed{\Delta \mid (A \wedge B)^F, \Gamma}} \xrightarrow{*} \cdots}\ (\wedge)$$

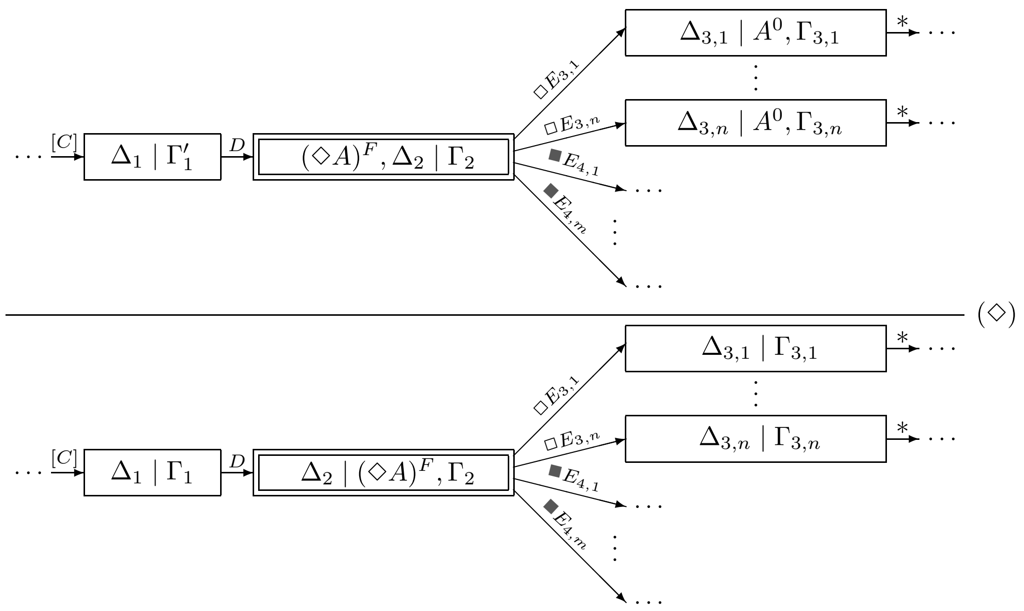
where: $\forall F' : (A \wedge B)^{F'} \notin \Delta$

$$\frac{\cdots \xrightarrow{[D]} \boxed{\boxed{(\Box A)^F, (\Diamond B_1)^{F_1}, \ldots, (\Diamond B_m)^{F_m}, \Delta \mid \Gamma}} \begin{array}{l} \xrightarrow{\Box A} \boxed{\epsilon \mid A^0, B_1{}^0, \ldots, B_m{}^0} \\ \xrightarrow{*} \cdots \end{array}}{\cdots \xrightarrow{[D]} \boxed{\boxed{(\Diamond B_1)^{F_1}, \ldots, (\Diamond B_m)^{F_m}, \Delta \mid (\Box A)^F, \Gamma}} \xrightarrow{*} \cdots}\ (\Box)$$

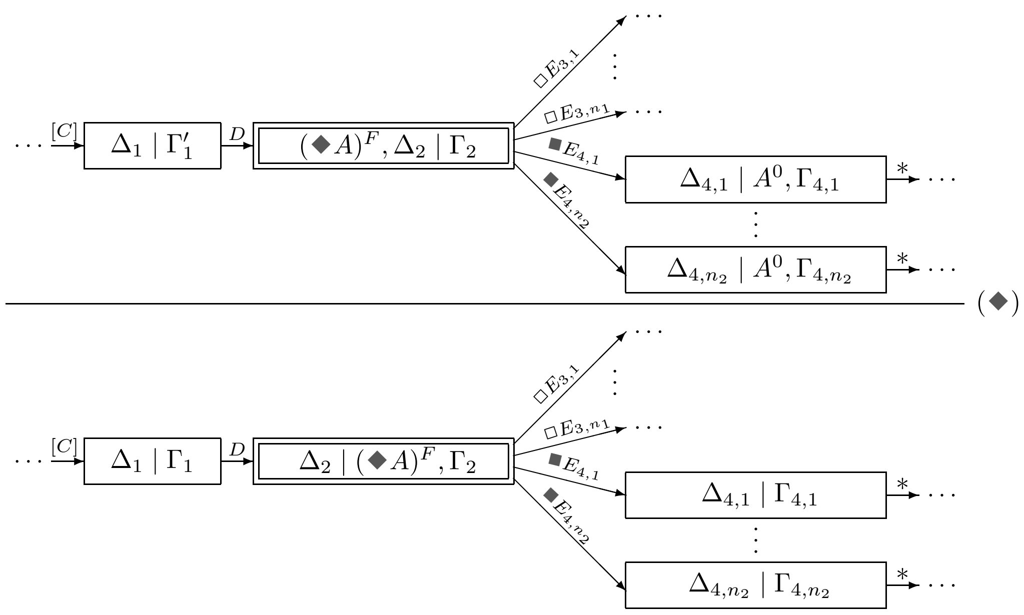
where: no $\Diamond$ fmls in Δ, and $\forall F' : (\Box A)^{F'} \notin \Delta$

$$\frac{\cdots \xrightarrow{[D]} \boxed{\boxed{(\blacksquare A)^F, (\blacklozenge B_1)^{F_1}, \ldots, (\blacklozenge B_m)^{F_m}, \Delta \mid \Gamma}} \begin{array}{l} \xrightarrow{\blacksquare A} \boxed{\epsilon \mid A^0, B_1{}^0, \ldots, B_m{}^0} \\ \xrightarrow{*} \cdots \end{array}}{\cdots \xrightarrow{[D]} \boxed{\boxed{(\blacklozenge B_1)^{F_1}, \ldots, (\blacklozenge B_m)^{F_m}, \Delta \mid (\blacksquare A)^F, \Gamma}} \xrightarrow{*} \cdots}\ (\blacksquare)$$

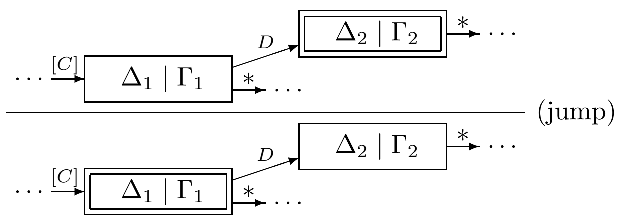
where: no $\blacklozenge$ fmls in Δ, and $\forall F' : (\blacksquare A)^{F'} \notin \Delta$

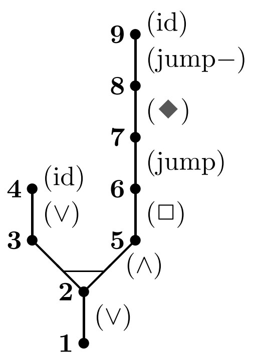

$\Delta_1 \mid \Gamma_1'$
$(\Diamond A)^F, \Delta_2 \mid \Gamma_2$
$\Delta_{3,1} \mid A^0, \Gamma_{3,1}$
$\Delta_{3,n} \mid A^0, \Gamma_{3,n}$
$\Delta_1 \mid \Gamma_1$
$\Delta_2 \mid (\Diamond A)^F, \Gamma_2$
$\Delta_{3,1} \mid \Gamma_{3,1}$
$\Delta_{3,n} \mid \Gamma_{3,n}$
$[C]$
D
$\Box E_{3,1}$
$\Box E_{3,n}$
$\blacksquare E_{4,1}$
$\blacklozenge E_{4,m}$
$*$
$(\Diamond)$

where: $\forall F' : (\Diamond A)^{F'} \notin \Delta_2$, and $\Gamma_1' = \begin{cases} \Gamma_1 & \text{if D is a } \Box\text{−formula} \\ A^D, \Gamma_1 & \text{if D is a } \blacksquare\text{−formula} \end{cases}$


$\Delta_1 \mid \Gamma_1'$
$(\blacklozenge A)^F, \Delta_2 \mid \Gamma_2$
$\Delta_{4,1} \mid A^0, \Gamma_{4,1}$
$\Delta_{4,n_2} \mid A^0, \Gamma_{4,n_2}$
$\Delta_1 \mid \Gamma_1$
$\Delta_2 \mid (\blacklozenge A)^F, \Gamma_2$
$\Delta_{4,1} \mid \Gamma_{4,1}$
$\Delta_{4,n_2} \mid \Gamma_{4,n_2}$
$[C]$
D
$\Box E_{3,1}$
$\Box E_{3,n_1}$
$\blacksquare E_{4,1}$
$\blacklozenge E_{4,n_2}$
$*$
$(\blacklozenge)$

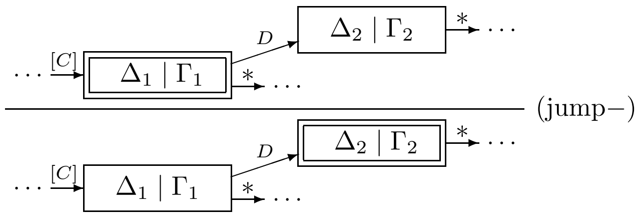
where: $\forall F' : (\blacklozenge A)^{F'} \notin \Delta_2$, and $\Gamma_1' = \begin{cases} \Gamma_1 & \text{if D is a } \blacksquare\text{−formula} \\ A^D, \Gamma_1 & \text{if D is a } \Box\text{−formula} \end{cases}$





The search starts at node 1.
The left branch of the branching at node 2 ends in an instance of the id) axiom (node 4).
On the right branch we apply (□) backwards and jump into the generated node. The backwards application of (◆) writes a $\neg p_0$ into the predecessor node. We jump back, and we find another instance of the (id) axiom (node 9). Thus $\boxed{\boxed{\epsilon \mid A^0}}$ is provable in $\mathsf{K}_t^{\mathcal{G},2}$.

Figure 5.q: The search tree in $\mathsf{K}_t^{\mathcal{G},2}$ of the example in remark 5.7.7. See figure 5.r for the corresponding proof.



5.7.6 THEOREM $\mathsf{K}_t^{\mathcal{G},2}$: invertible rules

All rules of $\mathsf{K}_t^{\mathcal{G},2}$ are invertible.

Proof

Analogous to the proof of theorem 5.2.4. Neither the labels of the arrows, nor those of the formulas are influenced by the order in which the rules are applied.

5.7.7 REMARK $\mathsf{K}_t^{\mathcal{G},2}$: backward proof search

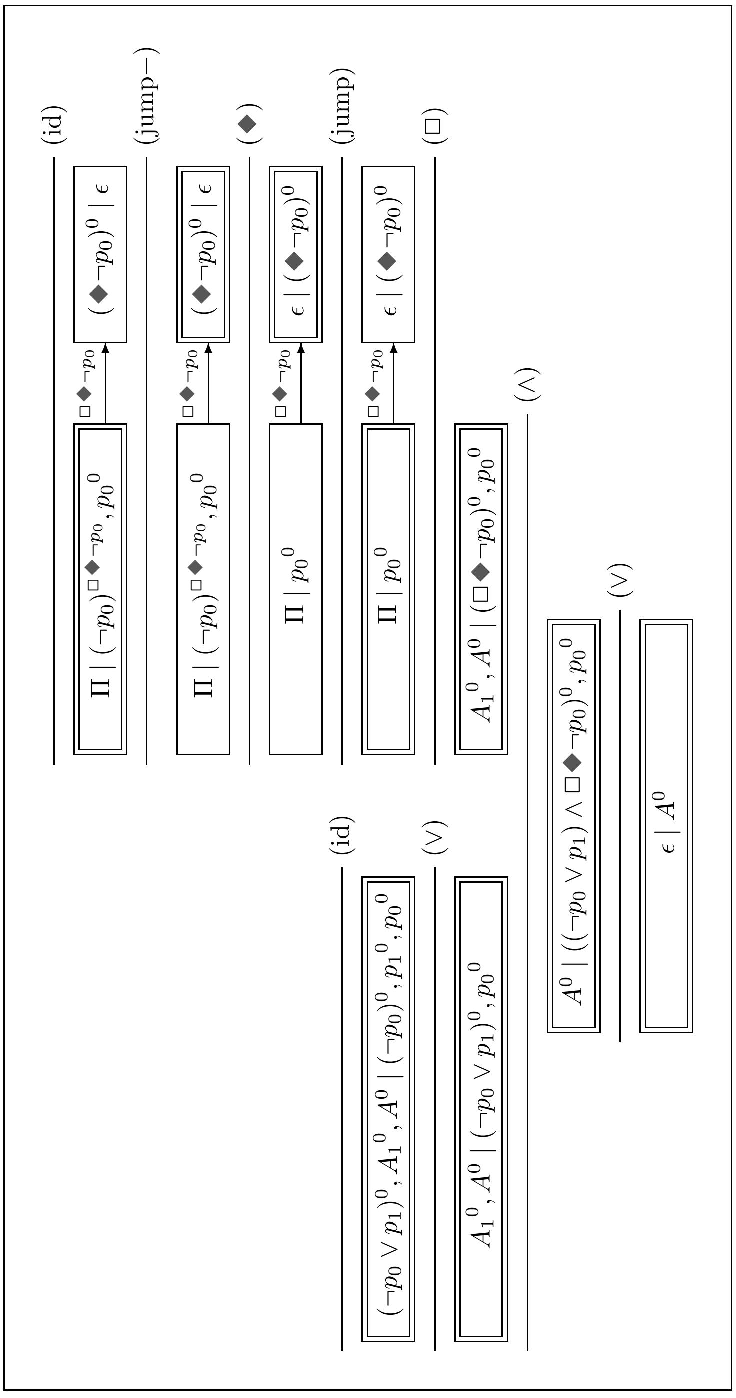
Since all rules are invertible, no backtracking is required when doing backward proof search in $\mathsf{K}_t^{\mathcal{G},2}$. In fact the search goes on exactly as in $\mathsf{K}_t^{\mathcal{G}}$.

Example

We check whether $\boxed{\boxed{\epsilon \mid A^0}}$ is provable in $\mathsf{K}_t^{\mathcal{G},2}$, where $A^0 \equiv (((\neg p_0 \vee p_1) \wedge \Box\blacklozenge\neg p_0) \vee p_0)^0$. We use the abbreviations ${A_1}^0 \equiv ((\neg p_0 \vee p_1) \wedge \Box\blacklozenge\neg p_0)^0$ and $\Pi = A^0, {A_1}^0, (\Box\blacklozenge\neg p_0)^0$. See figure 5.q for the search tree and figure 5.r for the corresponding proof. Compare these figures with the figures 4.u and 4.v from section 4.7.

5.7.8 THEOREM equivalence of $\mathsf{K}_t^{\mathcal{G}}$ and $\mathsf{K}_t^{\mathcal{G},2}$

$$\mathsf{K}_t^{\mathcal{G}} \vdash \boxed{\boxed{\epsilon \mid A}} \quad \Leftrightarrow \quad \mathsf{K}_t^{\mathcal{G},2} \vdash \boxed{\boxed{\epsilon \mid A^0}}$$



(id)
$(\neg p_0 \vee p_1)^0, {A_1}^0, {A_1}^0, A^0 \mid (\neg p_0)^0, {p_1}^0, {p_0}^0$
(∨)
${A_1}^0, A^0 \mid (\neg p_0 \vee p_1)^0, {p_0}^0$
(id)
$\Pi \mid (\neg p_0)^{\Box\blacklozenge\neg p_0}, {p_0}^0$
$\Box\blacklozenge\neg p_0$
$(\blacklozenge\neg p_0)^0 \mid \epsilon$
(jump−)
$\Pi \mid (\neg p_0)^{\Box\blacklozenge\neg p_0}, {p_0}^0$
$\Box\blacklozenge\neg p_0$
$(\blacklozenge\neg p_0)^0 \mid \epsilon$
(◆)
$\Pi \mid {p_0}^0$
$\Box\blacklozenge\neg p_0$
$\epsilon \mid (\blacklozenge\neg p_0)^0$
(jump)
$\Pi \mid {p_0}^0$
$\Box\blacklozenge\neg p_0$
$\epsilon \mid (\blacklozenge\neg p_0)^0$
(□)
${A_1}^0, A^0 \mid (\Box\blacklozenge\neg p_0)^0, {p_0}^0$
(∧)
$A^0 \mid ((\neg p_0 \vee p_1) \wedge \Box\blacklozenge\neg p_0)^0, {p_0}^0$
(∨)
$\epsilon \mid A^0$

Figure 5.r: The proof in $\mathsf{K}_t^{\mathcal{G},2}$ of the example in remark 5.7.7. See figure 5.q for the corresponding search tree.

Proof

'⇒':

Let $\mathcal{P}$ be a proof of $\boxed{\boxed{\epsilon \mid A}}$ in $\mathsf{K}_t^{\mathcal{G}}$. With an induction on formula length we can show that we can eliminate all steps $\frac{G_1}{G}$ in $\mathcal{P}$ where the main formula is an element of the left hand side of the current world of G. The resulting proof can be translated step by step (beginning at the axioms) into a proof of $\boxed{\boxed{\epsilon \mid A}}$ in $\mathsf{K}_t^{\mathcal{G},2}$.

'⇐':

Step by step we can translate proofs in $\mathsf{K}_t^{\mathcal{G},2}$ into proofs in $\mathsf{K}_t^{\mathcal{G}}$ (proof by induction on proof depth).

5.7.9 REMARK from $\mathsf{K}_t^{\mathcal{G},2}$ to $\mathsf{K}_t^{\mathcal{G},3}$

During backward proof search in $\mathsf{K}_t^{\mathcal{G},2}$, the rules ($\Diamond$) and ($\blacklozenge$) can put formulas into nodes that are nearer to the root than the actual node (see for example the example in remark 5.7.16). In order to apply rules backwards on these formulas we must have the possibility to jump towards the root. Therefore we cannot delete (jump−) from $\mathsf{K}_t^{\mathcal{G},2}$.

In $\mathsf{K}_t^{\mathcal{G},3}$ we try to foretell which formulas will be put into nodes that are nearer to the root. We do not know exactly what will happen later on in backward proof search, but we can give a finite number of possible sets of formulas that can be put in such a way into a certain node (function *indirect*). The rule (guess) helps us to go through these possibilities.

Some of the possibilities we investigate prove to be impossible, i.e. to be an instance of the ($\Box$clash) or ($\blacksquare$clash) axiom of $\mathsf{K}_t^{\mathcal{G},3}$. Others become instances of (id) axioms, and other branches fail.

5.7.10 DEFINITION *indirect*

First we define inductively the function *fmlat*.

1. $\mathit{fmlat}(P,0) := \{P\}$
   $\mathit{fmlat}(\neg P,0) := \{\neg P\}$
   $d \neq 0 \;\Rightarrow\; \mathit{fmlat}(P,d) := \emptyset,\ \mathit{fmlat}(\neg P,d) := \emptyset$
2. $\circ \in \{\wedge,\vee\} \;\Rightarrow\; \mathit{fmlat}(A\circ B,0) := \mathit{fmlat}(A,0)\cup \mathit{fmlat}(B,0)\cup\{A\circ B\}$
   $\circ \in \{\wedge,\vee\},\ d\neq 0 \;\Rightarrow\; \mathit{fmlat}(A\circ B,d) := \mathit{fmlat}(A,d)\cup \mathit{fmlat}(B,d)$
3. $\star \in \{\Box,\blacksquare\} \;\Rightarrow\; \mathit{fmlat}(\star A,0) := \mathit{fmlat}(A,-1)\cup\{\star A\}$
   $\star \in \{\Box,\blacksquare\},\ d\neq 0 \;\Rightarrow\; \mathit{fmlat}(\star A,d) := \mathit{fmlat}(A,d-1)$
4. $\star \in \{\Diamond,\blacklozenge\} \;\Rightarrow\; \mathit{fmlat}(\star A,0) := \mathit{fmlat}(A,-1)\cup \mathit{fmlat}(A,1)\cup\{\star A\}$
   $\star \in \{\Diamond,\blacklozenge\},\ d\neq 0 \;\Rightarrow\; \mathit{fmlat}(\star A,d) := \mathit{fmlat}(A,d-1)\cup \mathit{fmlat}(A,d+1)$

If Γ is the multiset $A_1,\ldots,A_n$, then $\mathit{fmlat}(\Gamma,d)$ stands for the set of formulas $\mathit{fmlat}(A_1,d)\cup\ldots\cup \mathit{fmlat}(A_n,d)$.

Now we can define the function *indirect*. In this definition, Δ' is the multiset Δ with the 'origins' of the formulas deleted, for example if $\Delta = (p_0\vee\neg p_1)^{\Box p_0}, (\Diamond p_1)^0$, then $\Delta' = p_0\vee\neg p_1, \Diamond p_1$.

$$\mathit{indirect}(\Delta) := \{A^{\Box B} \mid \blacklozenge A \in \mathit{fmlat}(\Delta',1), \Box B\in\Delta'\} \cup \{A^{\blacksquare B} \mid \Diamond A \in \mathit{fmlat}(\Delta',1), \blacksquare B\in\Delta'\}$$

Example

Assume that we do backward proof search in $\mathsf{K}_t^{\mathcal{G},2}$. If A occurs in a world w, then the set $\mathit{fmlat}(A,n)$ is a superset of the subformulas of A that can occur in a world with distance n from w (where $n>0$ means further away from the root).

$$\begin{aligned}
&\mathit{fmlat}(\blacksquare(p_3 \vee \Diamond(p_4 \wedge \blacklozenge p_1)), 1)\\
&\quad = \mathit{fmlat}(p_3 \vee \Diamond(p_4 \wedge \blacklozenge p_1), 0)\\
&\quad = \{p_3 \vee \Diamond(p_4 \wedge \blacklozenge p_1)\} \cup \mathit{fmlat}(p_3, 0) \cup \mathit{fmlat}(\Diamond(p_4 \wedge \blacklozenge p_1), 0)\\
&\quad = \{p_3 \vee \Diamond(p_4 \wedge \blacklozenge p_1), p_3, \Diamond(p_4 \wedge \blacklozenge p_1)\} \cup \mathit{fmlat}(p_4 \wedge \blacklozenge p_1, -1) \cup \mathit{fmlat}(p_4 \wedge \blacklozenge p_1, 1)\\
&\quad = \{p_3 \vee \Diamond(p_4 \wedge \blacklozenge p_1), p_3, \Diamond(p_4 \wedge \blacklozenge p_1)\}\\
&\qquad \cup \mathit{fmlat}(p_4, -1) \cup \mathit{fmlat}(\blacklozenge p_1, -1) \cup \mathit{fmlat}(p_4, 1) \cup \mathit{fmlat}(\blacklozenge p_1, 1)\\
&\quad = \{p_3 \vee \Diamond(p_4 \wedge \blacklozenge p_1), p_3, \Diamond(p_4 \wedge \blacklozenge p_1)\} \cup \mathit{fmlat}(p_1, -2) \cup \mathit{fmlat}(p_1, 0) \cup \mathit{fmlat}(p_1, 2)\\
&\quad = \{p_3 \vee \Diamond(p_4 \wedge \blacklozenge p_1), p_3, \Diamond(p_4 \wedge \blacklozenge p_1), p_1\}
\end{aligned}$$

$$\begin{aligned}
&\mathit{fmlat}(\Diamond\Diamond\Diamond\Diamond p_2, 1)\\
&\quad = \mathit{fmlat}(\Diamond\Diamond\Diamond p_2, 0) \cup \mathit{fmlat}(\Diamond\Diamond\Diamond p_2, 2)\\
&\quad = \{\Diamond\Diamond\Diamond p_2\} \cup \mathit{fmlat}(\Diamond\Diamond p_2, -1) \cup \mathit{fmlat}(\Diamond\Diamond p_2, 1) \cup \mathit{fmlat}(\Diamond\Diamond p_2, 3)\\
&\quad = \{\Diamond\Diamond\Diamond p_2\} \cup \mathit{fmlat}(\Diamond p_2, -2) \cup \mathit{fmlat}(\Diamond p_2, 0) \cup \mathit{fmlat}(\Diamond p_2, 2) \cup \mathit{fmlat}(\Diamond p_2, 4)\\
&\quad = \{\Diamond\Diamond\Diamond p_2, \Diamond p_2\} \cup \mathit{fmlat}(p_2, -3) \cup \mathit{fmlat}(p_2, -1) \cup \mathit{fmlat}(p_2, 1) \cup \mathit{fmlat}(p_2, 3) \cup \mathit{fmlat}(p_2, 5)\\
&\quad = \{\Diamond\Diamond\Diamond p_2, \Diamond p_2\}
\end{aligned}$$

$$\begin{aligned}
&\mathit{fmlat}(\blacksquare\blacksquare\neg p_2, 1)\\
&\quad = \mathit{fmlat}(\blacksquare\neg p_2, 0)\\
&\quad = \{\blacksquare\neg p_2\} \cup \mathit{fmlat}(\neg p_2, -1)\\
&\quad = \{\blacksquare\neg p_2\}
\end{aligned}$$

$$\begin{aligned}
&\mathit{indirect}((\blacksquare(p_3 \vee \Diamond(p_4 \wedge \blacklozenge p_1)))^{F_1}, (\Diamond\Diamond\Diamond\Diamond p_2)^{F_2}, (\blacksquare\blacksquare\neg p_2)^{F_3})\\
&\quad = \{(p_4 \wedge \blacklozenge p_1)^{\blacksquare(p_3 \vee \Diamond(p_4 \wedge \blacklozenge p_1))},\ (p_4 \wedge \blacklozenge p_1)^{\blacksquare\blacksquare\neg p_2},\\
&\qquad (\Diamond\Diamond p_2)^{\blacksquare(p_3 \vee \Diamond(p_4 \wedge \blacklozenge p_1))},\ (\Diamond\Diamond p_2)^{\blacksquare\blacksquare\neg p_2},\\
&\qquad {p_2}^{\blacksquare(p_3 \vee \Diamond(p_4 \wedge \blacklozenge p_1))},\ {p_2}^{\blacksquare\blacksquare\neg p_2}\}
\end{aligned}$$

5.7.11 THEOREM property of *indirect*

If we do backward proof search in $\mathsf{K}_t^{\mathcal{G},2}$ and the $(\blacklozenge)$ rule is applied backwards such that $\Gamma_1' = \Gamma_1, A^{\Box D}$, then $A^{\Box D} \in \mathit{indirect}(\Delta_1)$. If we do backward proof search in $\mathsf{K}_t^{\mathcal{G},2}$ and the $(\Diamond)$ rule is applied backwards such that $\Gamma_1' = \Gamma_1, A^{\blacksquare D}$, then $A^{\blacksquare D} \in \mathit{indirect}(\Delta_1)$. (See definition 5.7.5 for the meaning of Γ_1, Γ_1', $A^{\Box D}$, $A^{\blacksquare D}$, and Δ_1).

Proof

We prove only the first assertion, as the proof of the second one is analogous. Obviously $\Box D \in \Delta_1$. Hence it is sufficient to prove $\blacklozenge A \in \mathit{fmlat}(\Delta_1, 1)$.

Let w_1 be the world with Δ_1 and Γ_1, and let w_2 be the world with Δ_2 and $(\blacklozenge A)^F, \Gamma_2$ (cp. definition 5.7.5). At the beginning of backward proof search we had just one world w_0 with one formula C, and certainly $A \in \mathrm{subfmls}(C)$. There exists a 'path' X, starting at the root w_0 and ending in w_2, that shows how the formula $\blacklozenge A$ got from w_0 to w_2. This path can be seen as a tuple of pairs of formulas and worlds, i.e. $X = \langle\langle C, w_0\rangle, \ldots, \langle \blacklozenge A, w_2\rangle\rangle$.

As an example we consider the formula $(\neg p_0)^{\Box\blacklozenge\neg p_0}$ in the instance of (id) in the proof in figure 5.r. We call the root of the K_t graph w_0 and the second world w_2. Then the 'path' of the formula $(\neg p_0)^{\Box\blacklozenge\neg p_0}$ in w_0 is $\langle\langle A^0, w_0\rangle, \langle((\neg p_0 \vee p_1) \wedge \Box\blacklozenge\neg p_0)^0, w_0\rangle, \langle(\Box\blacklozenge\neg p_0)^0, w_0\rangle, \langle(\blacklozenge\neg p_0)^0, w_2\rangle,$ $\langle(\neg p_0)^{\Box\blacklozenge\neg p_0}, w_0\rangle\rangle$.

X must contain (maybe several times) the world w_1, i.e. $X = \langle\langle C, w_0\rangle, \ldots, \langle A_1, w_1\rangle, \ldots, \langle \blacklozenge A, w_2\rangle\rangle$. We look at that part of X from the last occurrence of w_1 in X to the end of X. We call this part Y, i.e. $Y = \langle\langle A_1, w_1\rangle, \ldots, \langle \blacklozenge A, w_2\rangle\rangle$ and w_1 occurs only in the first element of Y. From the definition of *fmlat* follows $\blacklozenge A \in \mathit{fmlat}(A_1, 1)$ (this is the only property of *fmlat* we need). Thus $\blacklozenge A \in \mathit{fmlat}(\Delta_1, 1)$.

5.7.12 REMARK variants of *indirect*

The definition of *indirect* is somewhat arbitrary. We could also define $\mathit{indirect}(\Delta) = \{A^{\Box B} \mid \Box B \in \Delta, \blacklozenge A \in \mathrm{subfmls}(\Delta)\} \cup \{A^{\blacksquare B} \mid \blacksquare B \in \Delta, \Diamond A \in \mathrm{subfmls}(\Delta)\}$ without losing the termination property of the calculus $\mathsf{K}_t^{\mathcal{G},3}$ defined below. The resulting calculus would still be equivalent to $\mathsf{K}_t^{\mathcal{G},2}$. We hope that our definition of *indirect* helps to motivate the calculus.

Note that $\mathit{indirect}(\Delta) = \{A^{\Box B} \mid \Box B \in \Delta \text{ and } A \in \mathrm{subfmls}(E)\} \cup \{A^{\blacksquare B} \mid \blacksquare B \in \Delta \text{ and } A \in \mathrm{subfmls}(E)\}$, where E is the formula for which we do backward proof search, is not a suitable alternative, since then backward proof search in $\mathsf{K}_t^{\mathcal{G},3}$ would no longer terminate.

5.7.13 DEFINITION graph calculus $\mathsf{K}_t^{\mathcal{G},3}$

axioms:

$$\frac{}{\cdots \xrightarrow{[D]} \boxed{\boxed{\Pi \mid \Delta \mid \mathsf{true}^F, \Gamma}} \xrightarrow{*} \cdots}\ (\mathsf{true})$$

$$\frac{}{\cdots \xrightarrow{[D]} \boxed{\boxed{\Pi \mid \Delta \mid P^{F_1}, (\neg P)^{F_2}, \Gamma}} \xrightarrow{*} \cdots}\ (\mathrm{id})$$

$$\begin{tikzcd}
 & & \boxed{\Pi_1 \mid \Delta_1 \mid \Gamma_1} \arrow[r, "*"] & \cdots \\
 & & \vdots & \\
 & & \boxed{\Pi_n \mid \Delta_n \mid \Gamma_n} \arrow[r, "*"] & \cdots \\
\cdots \arrow[r] & \boxed{\boxed{\Pi \mid \Delta \mid \Gamma}} \arrow[uuur, "\Box D_1"] \arrow[ur, "\Box D_n"] \arrow[dr, "\blacksquare E_1"] \arrow[dddr, "\blacklozenge E_m"] & & & (\Box\mathrm{clash}) \\
 & & \cdots & \\
 & & \vdots & \\
 & & \cdots &
\end{tikzcd}$$

where: no other rule is applicable backwards even after auxiliary backward applications of (jump) and (jump−), and there is an $i \in \{1, \ldots, n\}$ such that $\{A' \mid (A')^{\Box D_i} \in \Gamma, \Delta\} \neq \{B' \mid \exists F' : (\blacklozenge B')^{F'} \in \Delta_i\}$

$$\begin{tikzcd}
 & & \cdots & \\
 & & \vdots & \\
 & & \cdots & \\
\cdots \arrow[r] & \boxed{\boxed{\Pi \mid \Delta \mid \Gamma}} \arrow[uuur, "\Box D_1"] \arrow[ur, "\Box D_n"] \arrow[dr, "\blacksquare E_1"] \arrow[dddr, "\blacklozenge E_m"] & & & (\blacksquare\mathrm{clash}) \\
 & & \boxed{\Pi_1 \mid \Delta_1 \mid \Gamma_1} \arrow[r, "*"] & \cdots \\
 & & \vdots & \\
 & & \boxed{\Pi_m \mid \Delta_m \mid \Gamma_m} \arrow[r, "*"] & \cdots
\end{tikzcd}$$

where: no other rule is applicable backwards even after auxiliary backward applications of (jump) and (jump−), and there is an $i \in \{1, \ldots, m\}$ such that $\{A' \mid (A')^{\blacksquare E_i} \in \Gamma, \Delta\} \neq \{B' \mid \exists F' : (\Diamond B')^{F'} \in \Delta_i\}$

$$\frac{\cdots \xrightarrow{[D]} \boxed{\boxed{A^F, \Pi \mid \Delta \mid A^F, \Gamma}} \xrightarrow{*} \cdots \qquad \cdots \xrightarrow{[D]} \boxed{\boxed{A^F, \Pi \mid \Delta \mid \Gamma}} \xrightarrow{*} \cdots}{\cdots \xrightarrow{[D]} \boxed{\boxed{\Pi \mid \Delta \mid \Gamma}} \xrightarrow{*} \cdots}\ (\mathrm{guess})$$

where: $A^F \in \mathit{indirect}(\Delta)$ and $A^F \notin \Pi$

We obtain the rules (∨), (∧), (□), (■), (jump), (jump−) of $\mathsf{K}_t^{\mathcal{G},3}$ in the obvious way by adding appropriate third components to the corresponding rules of $\mathsf{K}_t^{\mathcal{G},2}$:

- (∨), (∧): Add Π to the actual nodes. In the same way we could construct the axioms (id) and (true) of $\mathsf{K}_t^{\mathcal{G},3}$ from the axioms of $\mathsf{K}_t^{\mathcal{G},2}$.
- (□), (■): Add Π to the actual nodes, and ϵ to the generated nodes.
- (jump), (jump−): Add Π_1 to the left node and Π_2 to the right node.

We obtain also the rules (◇) and (◆) of $\mathsf{K}_t^{\mathcal{G},3}$ from the corresponding rules of $\mathsf{K}_t^{\mathcal{G},2}$:

- Add Π_1 to the nodes with Δ_1, add Π_2 to the nodes with Δ_2, add $\Pi_{1,1}$ to the nodes with $\Delta_{1,1}$. …add Π_{4,n_2} to the nodes with Δ_{4,n_2}.
- Replace Γ_1' by Γ_1.

5.7.14 REMARK (◇) and (◆)

The essential change from $\mathsf{K}_t^{\mathcal{G},2}$ to $\mathsf{K}_t^{\mathcal{G},3}$ is the replacement of Γ_1' by Γ_1 in the two rules (◇) and (◆): The new rules (◇) and (◆) no longer put any formulas into worlds that are nearer to the root. Therefore it becomes possible to eliminate (jump−).

5.7.15 THEOREM $\mathsf{K}_t^{\mathcal{G},3}$: invertible rules

All rules of $\mathsf{K}_t^{\mathcal{G},3}$ are invertible.

Proof

Analogous to the corresponding proof for $\mathsf{K}_t^{\mathcal{G},2}$ (theorem 5.2.4). Note that it is not possible that both (□clash) and for example (∨) are applicable backwards because of the condition in (□clash).

5.7.16 REMARK $\mathsf{K}_t^{\mathcal{G},3}$: backward proof search

Compared to backward proof search in $\mathsf{K}_t^{\mathcal{G},2}$, we have a new rule (guess) and we have new axioms (□clash) and (■clash). If we reach a (□clash) or (■clash) axiom, then this means that previous applications of (guess) have led to an inconsistent situation.

Example

We check whether $\boxed{\boxed{\epsilon \mid \epsilon \mid A^0}}$ is provable in $\mathsf{K}_t^{\mathcal{G},3}$, where $A^0 \equiv (((\neg p_0 \vee p_1) \wedge \Box\blacklozenge\neg p_0) \vee p_0)^0$. We use the abbreviations ${A_1}^0 \equiv (\neg p_0 \vee p_1) \wedge \Box\blacklozenge\neg p_0$, and $\Pi = (\Box\Diamond\neg p_0)^0, {A_1}^0, A^0$. See figure 5.s for the search tree and figure 5.t for the corresponding proof.

5.7.17 THEOREM equivalence of $\mathsf{K}_t^{\mathcal{G},2}$ and $\mathsf{K}_t^{\mathcal{G},3}$

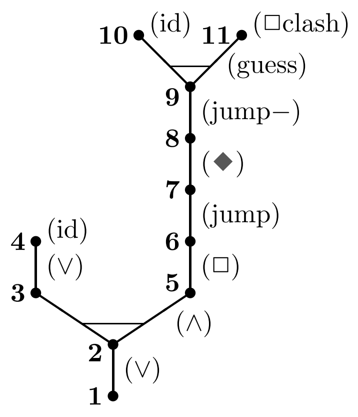
$$\mathsf{K}_t^{\mathcal{G},2} \vdash \boxed{\boxed{\epsilon \mid A^0}} \quad \Leftrightarrow \quad \mathsf{K}_t^{\mathcal{G},3} \vdash \boxed{\boxed{\epsilon \mid \epsilon \mid A^0}}$$

Proof

We do backward proof search in $\mathsf{K}_t^{\mathcal{G},2}$, starting with $\boxed{\boxed{\epsilon \mid A^0}}$. The result is a complete $\mathsf{K}_t^{\mathcal{G},2}$ search tree. (We restrict (jump) and (jump−) as in the case of $\mathsf{K}^{\mathcal{S}}$ in order to avoid infinite series of backward applications of these rules.)

We simulate this proof search in $\mathsf{K}_t^{\mathcal{G},3}$, starting with $\boxed{\boxed{\epsilon \mid \epsilon \mid A^0}}$.

First we replace applications of rules of $\mathsf{K}_t^{\mathcal{G},2}$ by applications of rules of $\mathsf{K}_t^{\mathcal{G},3}$ as follows:



The search starts at node 1. The conjunction causes a branching at node 2. The left branch ends in instance of an axiom (node 4). On the right branch at node 9 we can only apply (guess). This leads to instances of the (id) and the (□clash) axiom (nodes 10 and 11).
Thus $\boxed{\boxed{\epsilon \mid \epsilon \mid A^0}}$ is provable in $\mathsf{K}_t^{\mathcal{G},3}$. Compare this search tree with the one in figure 5.q.

Figure 5.s: The search tree in $\mathsf{K}_t^{\mathcal{G},3}$ of the example in remark 5.7.16. See 5.t for the corresponding proof.

- (id) $\rightsquigarrow$ (id)
- (∨) $\rightsquigarrow$ (∨), (∧) $\rightsquigarrow$ (∧)
- (□) $\rightsquigarrow$ (□), (■) $\rightsquigarrow$ (■)
- (jump) $\rightsquigarrow$ (jump) (we choose 'the same' world to jump into)
- (jump−) $\rightsquigarrow$ (jump−) (we choose 'the same' world to jump into)
- (◇) $\rightsquigarrow$ (◇) if the label D (in the rule of $\mathsf{K}_t^{\mathcal{G},2}$) is a ■ formula
- (◆) $\rightsquigarrow$ (◆) if the label D (in the rule of $\mathsf{K}_t^{\mathcal{G},2}$) is a □ formula

There remain only two interesting cases: (◇) where the label D is a □ formula, and the analogous case for (◆). We simulate such a (◇) application as follows in $\mathsf{K}_t^{\mathcal{G},3}$:

- Apply (◇) backwards.
- Jump with (jump−) into the predecessor node.
- Apply (guess) backwards, with main formula A^F. This is always possible because of theorem 5.7.11.
- On both branches of the search tree generated by the backward application of (guess) we use (jump) to get back.

Afterwards we continue simulating the proof search in the left branch of the search tree (i.e. in the branch where the backward application of (guess) added a formula). We simulate the corresponding (◆) application analogously with (◆), (jump−), (guess), (jump). After this transformation we have an incomplete $\mathsf{K}_t^{\mathcal{G},3}$ search tree for $\boxed{\boxed{\epsilon \mid \epsilon \mid A^0}}$.

In order to obtain a complete $\mathsf{K}_t^{\mathcal{G},3}$ search tree, we have to complete three sorts of nodes of this tree:

1. The second premise that has been generated when translating (◆) with D being a □ formula.
2. The second premise that has been generated when translating (◇) with D being a ■ formula.
3. Leaves of the search tree in $\mathsf{K}_t^{\mathcal{G},2}$ that are not instances of an axiom have been transformed into nodes on which (guess) is still applicable.

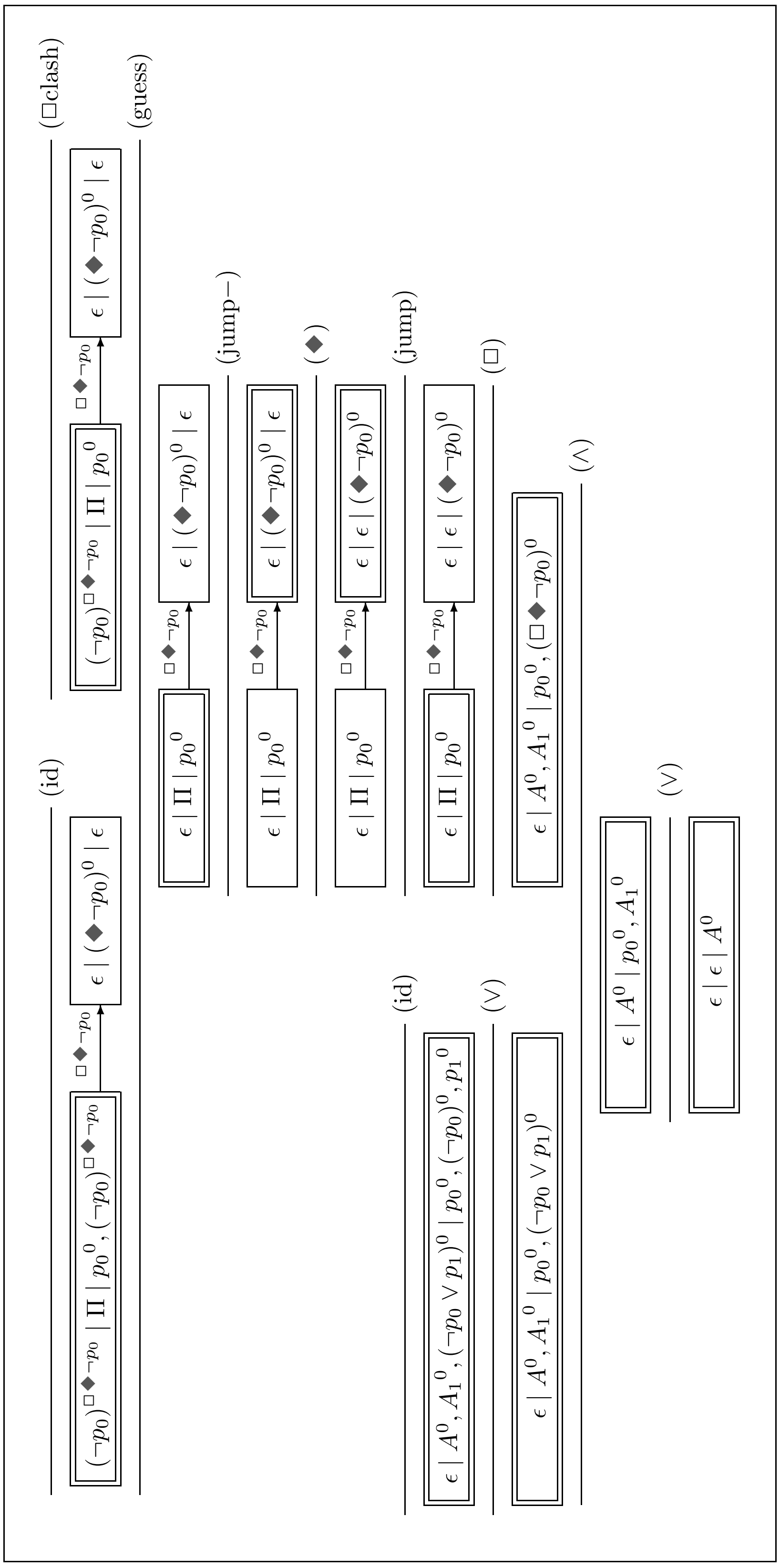


Figure 5.t: The proof in $\mathsf{K}_t^{\mathcal{G},3}$ of the example in remark 5.7.16. See 5.s for the corresponding search tree.

We continue with backward proof search in $\mathsf{K}_t^{\mathcal{G},2}$ at all these nodes. Let N be such a node, and T the tree with root N that was generated when completing the node N.

If N is of the form 1., then all the leaves of T become instances of the ($\Box$clash) axiom of $\mathsf{K}_t^{\mathcal{G},3}$. Proof: The K_t graph at the node N has the form

$$\begin{tikzcd}
\cdots \arrow[r] & \boxed{B^{\Box D}, \Pi_1 \mid \Box D, \Delta_1 \mid \Gamma_1} \arrow[r, "\Box D"] \arrow[dr, "*"] & \boxed{\Pi_2 \mid (\blacklozenge B)^F, \Delta_2 \mid \Gamma_2} \arrow[r, "*"] & \cdots \\
 & & \cdots &
\end{tikzcd}$$

where $B^{\Box D} \notin \Gamma_1$. In general this is not an instance of the ($\Box$clash) axiom: It is possible that, perhaps after suitable applications of (jump) and (jump$-$), other rules are applicable backwards. Assume now that one of the leaves of T is not an instance of an axiom. Then it is also not an instance of the ($\Box$clash) axiom. This means that during the backward proof search which starts at N and leads to this leaf of T, the formula $B^{\Box D}$ has been added to Γ_1. Thus $(\blacklozenge B)^{F'}$ has been added to Δ_2. This is not possible because of the condition in the ($\blacklozenge$) rule.

If N is of the form 2., then all the leaves of T become instances of the ($\blacksquare$clash) axiom of $\mathsf{K}_t^{\mathcal{G},3}$. The proof is analogous to the case where N is of the form 1.

If N is of the form 3., then the leaf of the rightmost branch of T is not an axiom of $\mathsf{K}_t^{\mathcal{G},3}$, since we added formulas only to the leftmost sequent of this leaf (cp. the rule (guess)).

Thus we have:

$$\mathsf{K}_t^{\mathcal{G},2} \vdash \boxed{\boxed{\epsilon \mid A^0}} \quad \Rightarrow \quad \mathsf{K}_t^{\mathcal{G},3} \vdash \boxed{\boxed{\epsilon \mid \epsilon \mid A^0}}$$
$$\mathsf{K}_t^{\mathcal{G},2} \nvdash \boxed{\boxed{\epsilon \mid A^0}} \quad \Rightarrow \quad \mathsf{K}_t^{\mathcal{G},3} \nvdash \boxed{\boxed{\epsilon \mid \epsilon \mid A^0}}$$

5.7.18 REMARK from $\mathsf{K}_t^{\mathcal{G},3}$ to $\mathsf{K}_t^{\mathcal{S}}$

The step from $\mathsf{K}_t^{\mathcal{G},3}$ to $\mathsf{K}_t^{\mathcal{S}}$ is about the same as the one from $\mathsf{K}^{\mathcal{G}}$ to $\mathsf{K}^{\mathcal{S}}$. The rules (jump)/(jump$-$) and ($\Box$) of $\mathsf{K}_t^{\mathcal{G},3}$ are combined in the rule ($\Box$) of $\mathsf{K}_t^{\mathcal{S}}$, and the rules (jump)/(jump$-$) and ($\blacksquare$) of $\mathsf{K}_t^{\mathcal{G},3}$ are combined in the rule ($\blacksquare$) of $\mathsf{K}_t^{\mathcal{S}}$.

For convenience we still draw boxes and arrows, but obviously we could just as well write $\Lambda \mid D \mid \Pi \mid \Delta \mid \Gamma$ instead of $\begin{tikzcd} \boxed{\Lambda} \arrow[r, dashed, "D"] & \boxed{\Pi \mid \Delta \mid \Gamma} \end{tikzcd}$.

Note that the rules ($\Box$) and ($\blacksquare$) of $\mathsf{K}_t^{\mathcal{S}}$ are not invertible. Thus we have to use backtracking during the backward proof search in $\mathsf{K}_t^{\mathcal{S}}$ (as for example for $\mathsf{K}^{\mathcal{S}}$).

5.7.19 REMARK $\mathsf{K}_t^{\mathcal{S}}$: size of sequents

If we do backward proof search for $\boxed{\epsilon \mid \epsilon \mid A^0}$ in $\mathsf{K}_t^{\mathcal{S}}$, then all sequents that occur have a size that is polynomial in length(A), whereas the graphs of $\mathsf{K}_t^{\mathcal{G},3}$ can have exponential size. (See also chapter 6.) Note that the sequents in the sequent calculus for K_t in [Kas94] can have exponential size if we do backward proof search.

5.7.20 DEFINITION sequent calculus $\mathsf{K}_t^{\mathcal{S}}$

We use the following convention: $\begin{tikzcd} \boxed{\Lambda} \arrow[r, dashed, "D"] & \boxed{\Pi \mid \Delta \mid \Gamma} \end{tikzcd}$ means that the world $\boxed{\Pi \mid \Delta \mid \Gamma}$ can have a predecessor (with content Λ and label D on the connecting arrow), but it needs not to. Without this convention we would have to write down two versions for all axioms and all rules except ($\Box$clash) and ($\blacksquare$clash).

axioms:

$$\frac{}{\boxed{\Lambda} \overset{D}{\dashrightarrow} \boxed{\Pi \mid \Delta \mid \mathsf{true}, \Gamma}}\ (\mathsf{true})$$

$$\frac{}{\boxed{\Lambda} \overset{D}{\dashrightarrow} \boxed{\Pi \mid \Delta \mid P, \neg P, \Gamma}}\ (\mathrm{id})$$

$$\frac{}{\boxed{\Lambda} \xrightarrow{\Box D} \boxed{\Pi \mid \Delta \mid \Gamma}}\ (\Box\mathrm{clash})$$

where: $(\vee)$, $(\wedge)$, $(\Diamond)$, $(\blacklozenge)$, and (guess) not applicable backwards,
$\{A' \mid (A')^{\Box D} \in \Lambda\} \neq \{B' \mid \exists F' : (\blacklozenge B')^{F'} \in \Delta\}$

$$\frac{}{\boxed{\Lambda} \xrightarrow{\blacksquare D} \boxed{\Pi \mid \Delta \mid \Gamma}}\ (\blacksquare\mathrm{clash})$$

where: $(\vee)$, $(\wedge)$, $(\Diamond)$, $(\blacklozenge)$, and (guess) not applicable backwards,
$\{A' \mid (A')^{\blacksquare D} \in \Lambda\} \neq \{B' \mid \exists F' : (\Diamond B')^{F'} \in \Delta\}$

rules:

$$\frac{\boxed{\Lambda} \overset{D}{\dashrightarrow} \boxed{\Pi \mid (A \vee B)^F, \Delta \mid A^0, B^0, \Gamma}}{\boxed{\Lambda} \overset{D}{\dashrightarrow} \boxed{\Pi \mid \Delta \mid (A \vee B)^F, \Gamma}}\ (\vee)$$

where: $\forall F' : (A \vee B)^{F'} \notin \Delta$

$$\frac{\boxed{\Lambda} \overset{D}{\dashrightarrow} \boxed{\Pi \mid (A \wedge B)^F, \Delta \mid A^0, \Gamma} \qquad \boxed{\Lambda} \overset{D}{\dashrightarrow} \boxed{\Pi \mid (A \wedge B)^F, \Delta \mid B^0, \Gamma}}{\boxed{\Lambda} \overset{D}{\dashrightarrow} \boxed{\Pi \mid \Delta \mid (A \wedge B)^F, \Gamma}}\ (\wedge)$$

where: $\forall F' : (A \wedge B)^{F'} \notin \Delta$

$$\frac{\boxed{\Lambda} \overset{D}{\dashrightarrow} \boxed{\Pi \mid (\Diamond A)^F, \Delta \mid \Gamma}}{\boxed{\Lambda} \overset{D}{\dashrightarrow} \boxed{\Pi \mid \Delta \mid (\Diamond A)^F, \Gamma}}\ (\Diamond)$$

where: $\forall F' : (\Diamond A)^{F'} \notin \Delta$

$$\frac{\boxed{\Lambda} \overset{D}{\dashrightarrow} \boxed{\Pi \mid (\blacklozenge A)^F, \Delta \mid \Gamma}}{\boxed{\Lambda} \overset{D}{\dashrightarrow} \boxed{\Pi \mid \Delta \mid (\blacklozenge A)^F, \Gamma}}\ (\blacklozenge)$$

where: $\forall F' : (\blacklozenge A)^{F'} \notin \Delta$

$$\frac{\boxed{\Lambda} \dashrightarrow^{D} \boxed{A^F, \Pi \mid \Delta \mid A^F, \Gamma} \qquad \boxed{\Lambda} \dashrightarrow^{D} \boxed{A^F, \Pi \mid \Delta \mid \Gamma}}{\boxed{\Lambda} \dashrightarrow^{D} \boxed{\Pi \mid \Delta \mid \Gamma}} \;(\text{guess})$$

where: $A^F \in \mathit{indirect}(\Delta)$, $A^F \notin \Pi$

$$\frac{\boxed{(\Diamond A_1)^{F_1}, \ldots, (\Diamond A_m)^{F_m}, \Delta, (\Box B)^F, \Gamma} \xrightarrow{\Box B} \boxed{\epsilon \mid \epsilon \mid B^0, {A_1}^0, \ldots, {A_m}^0}}{\boxed{\Lambda} \dashrightarrow^{D} \boxed{\Pi \mid (\Diamond A_1)^{F_1}, \ldots, (\Diamond A_m)^{F_m}, \Delta \mid (\Box B)^F, \Gamma}} \;(\Box)$$

where: no $\Diamond$ formulas in Δ, and $(\vee)$, $(\wedge)$, $(\Diamond)$, $(\blacklozenge)$, (guess) not applicable backwards

$$\frac{\boxed{(\blacklozenge A_1)^{F_1}, \ldots, (\blacklozenge A_m)^{F_m}, \Delta, (\blacksquare B)^F, \Gamma} \xrightarrow{\blacksquare B} \boxed{\epsilon \mid \epsilon \mid B^0, {A_1}^0, \ldots, {A_m}^0}}{\boxed{\Lambda} \dashrightarrow^{D} \boxed{\Pi \mid (\blacklozenge A_1)^{F_1}, \ldots, (\blacklozenge A_m)^{F_m}, \Delta \mid (\blacksquare B)^F, \Gamma}} \;(\blacksquare)$$

where: no $\blacklozenge$ formulas in Δ, and $(\vee)$, $(\wedge)$, $(\Diamond)$, $(\blacklozenge)$, (guess) not applicable backwards

5.7.21 THEOREM equivalence of $\mathsf{K}_t^{\mathcal{G},3}$ and $\mathsf{K}_t^{\mathcal{S}}$

$$\mathsf{K}_t^{\mathcal{G},3} \vdash \boxed{\boxed{\epsilon \mid \epsilon \mid A^0}} \quad \Leftrightarrow \quad \mathsf{K}_t^{\mathcal{S}} \vdash \boxed{\epsilon \mid \epsilon \mid A^0}$$

Proof

'$\Leftarrow$'

We can translate the proofs in $\mathsf{K}_t^{\mathcal{S}}$ stepwise into proofs in $\mathsf{K}_t^{\mathcal{G},3}$. Only the rules ($\Box$clash) and ($\blacksquare$clash) are a bit more difficult.

'$\Leftarrow$'

Since all rules of $\mathsf{K}_t^{\mathcal{G},3}$ are invertible, we can rearrange the proof in $\mathsf{K}_t^{\mathcal{G},3}$ such that it can be translated stepwise into a proof in $\mathsf{K}_t^{\mathcal{S}}$. We used the same method in the proof of theorem 5.2.14 where we proved the equivalence of $\mathsf{K}^{\mathcal{S}}$ and $\mathsf{K}^{\mathcal{S},2}$.

5.7.22 THEOREM $\mathsf{K}_t^{\mathcal{S}}$: termination

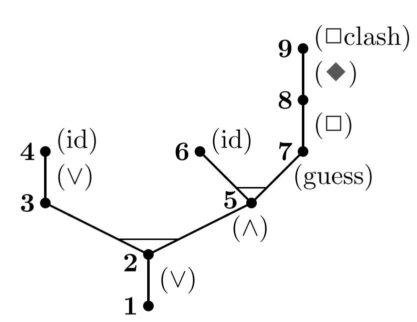
Backward proof search in $\mathsf{K}_t^{\mathcal{S}}$ terminates.

Proof

Assume that we do backward proof search for $\boxed{\epsilon \mid \epsilon \mid A^0}$.

For termination, it is essential that each formula that moves from a world to a child of this world becomes shorter, and that card(indirect(subfmls(C))) $\leq$ (length(C))^2. We now define $m(\boxed{\Lambda} \dashrightarrow^{D} \boxed{\Pi \mid \Delta \mid \Gamma}) := (c^4 + c^2) \cdot \max(\{\text{length}(B) \mid B \in \Delta, \Gamma\}) + c^2 \cdot (c^2 - \text{card}(\Pi)) + \text{length}(\Gamma)$, where $c := \text{length}(A)$.

For all K_t graphs G that can occur during backward proof search we have $m(G) \geq 0$. Obviously the value decreases if we apply one of the rules $(\wedge)$, $(\vee)$, $(\Diamond)$, $(\blacklozenge)$, (guess) backwards. The interesting cases are the rules $(\Box)$ and $(\blacksquare)$. If G_1 is the premise and G_2 is the conclusion of an application of



The search starts at node 1. Since backtracking is not necessary (nor possible), we obtain essentially the same search tree as in figure 5.s. There are two differences: We have to apply (guess) earlier, and the (jump) rule is built-in in the (□) rule.

Figure 5.u: The search tree in $\mathsf{K}_t^{\mathcal{S}}$ of the example in remark 5.7.23. See figure 5.v for the corresponding proof.

one of these rules, then we have: $m(G_1)-m(G_2) > (c^4+c^2)\cdot 1+c^2\cdot((c^2-c^2)-(c^2-0))+(0-c^2) = c^4+c^2-(c^4+c^2) = 0$.

5.7.23 REMARK $\mathsf{K}_t^{\mathcal{S}}$: backward proof search

In contrast to backward proof search in $\mathsf{K}_t^{\mathcal{G},3}$ we have to backtrack when a branch fails. As far as backward proof search is concerned, the difference between $\mathsf{K}_t^{\mathcal{G},3}$ and $\mathsf{K}_t^{\mathcal{S}}$ is the same as between in $\mathsf{K}^{\mathcal{G},2}$ and in $\mathsf{K}^{\mathcal{S}}$.

Example

We check whether $\boxed{\epsilon \mid \epsilon \mid A^0}$ is provable in $\mathsf{K}_t^{\mathcal{S}}$, where $A^0 \equiv (((\neg p_0 \vee p_1) \wedge \Box\blacklozenge\neg p_0) \vee p_0)^0$. We use the abbreviation ${A_1}^0 \equiv ((\neg p_0 \vee p_1) \wedge \Box\blacklozenge\neg p_0)^0$. See figure 5.u for the search tree and 5.v for the corresponding proof.

5.8 OTHER LOGICS

5.8.1 REMARK other propositional modal logics

Sequent calculi for many other logics are discussed in [Fit83] and [Gor]. In addition, there are many papers concerning tableaux and sequent calculi for modal logics (see the introduction).

5.8.2 REMARK K_n, $\mathsf{K}_n + T$, KT_n, $\mathsf{KT}_n + T$, $\mathsf{S4}_n$, $\mathsf{S4}_n + T$

Sequent calculi for K_n, $\mathsf{K}_n + T$, KT_n, and $\mathsf{KT}_n + T$ follow immediately from the corresponding sequent calculi for K, $\mathsf{K} + T$, KT, and $\mathsf{KT} + T$. The (□) rule for K_n then looks as follows:

$$\frac{A, \Gamma}{\Box_i A, \Diamond_i \Gamma, \Lambda}\ (\Box)$$

i.e. the $\Diamond$ formulas with the ‘wrong’ index are removed during a backward application of (□).

In contrast to $\mathsf{S4}$, in $\mathsf{S4}_n$ a $\Diamond_i$ formula can disappear during backward proof search. Therefore the history is not just a multiset of formulas, but a multiset of multisets of formulas of $\Box_i$ and

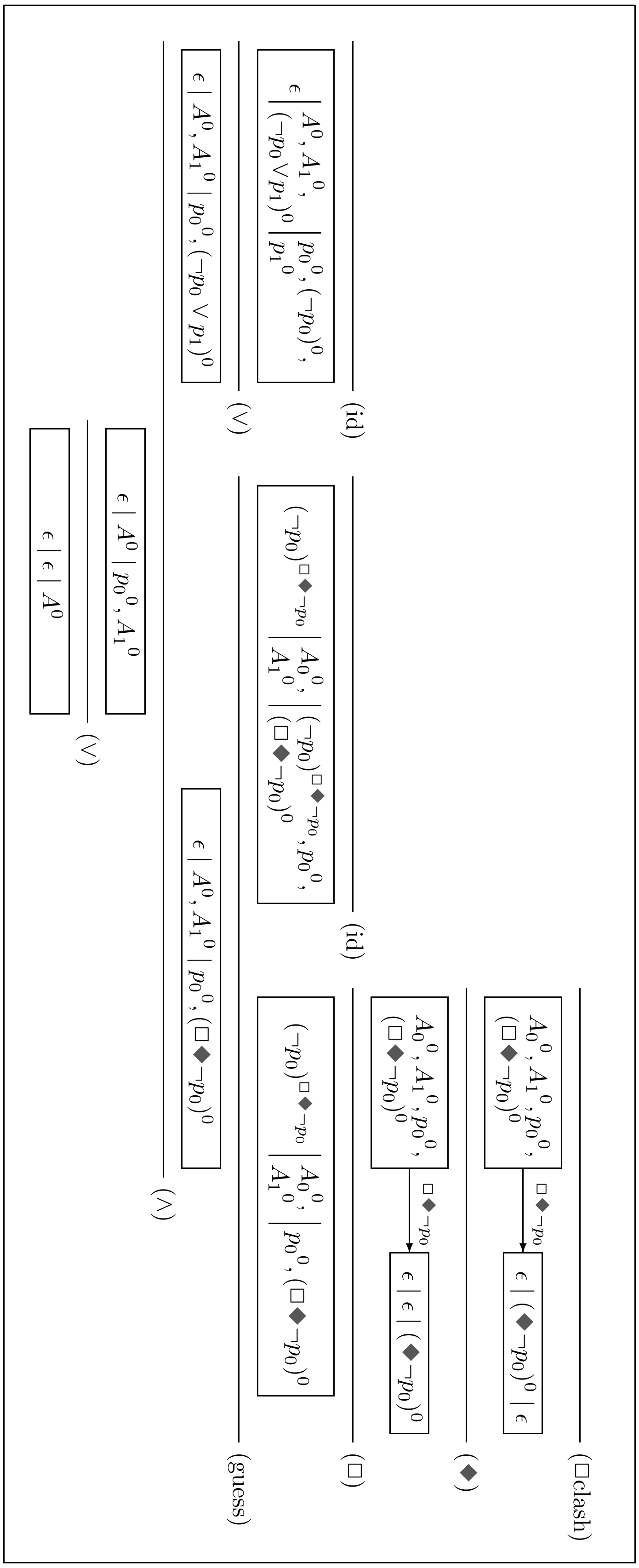


Figure 5.v: The proof in K_t^S of the example in remark 5.7.23. See figure 5.u for the corresponding search tree.

$\Diamond_i$ formulas (as for $\mathsf{K}+T$ for example). Moreover, it is not possible to embed $\mathsf{S4}_n+T$ in $\mathsf{S4}_n$ as we could embed $\mathsf{S4}+T$ in $\mathsf{S4}$.

The logics K_n+T, KT_n+T, $\mathsf{S4}_n+T$ can be used to solve the so-called wise men puzzle (see for example [Fit93]).

5.8.3 REMARK CPC

The usual one-sided sequent calculus for CPC is the calculus $\mathsf{K}^{\mathcal{S},2}$ without the ($\Box$) rule.

5.8.4 REMARK IPC

Sequent calculi for backward proof search in IPC are discussed in [Dyc92], [SFH92], and [TS96]. In contrast to S4, it is possible to do without a hidden contraction. Sequent calculi with loop-check are discussed in [HSZ96], and [How97]. See also [Ott97] and [Tam96].

5.8.5 REMARK many-valued logics

Proof search in sequent calculi for many-valued logics is discussed in [Häh93].

5.8.6 REMARK linear logic

In [Tam94] backward proof search in sequent calculi for linear logic is discussed thoroughly. See also [And92] and [dG95].

5.9 SUMMARY

Beginning with the graph calculi, we have developed sequent calculi for the logics K, $\mathsf{K}+T$, KT, $\mathsf{KT}+T$, and $\mathsf{S4}$. The equivalence with the graph calculi has been proved constructively, and we have shown how to obtain Hilbert-style proofs from sequent calculus proofs. With numerous examples we have exposed the connections between possible world semantics, graph calculi and sequent calculi. This includes the extraction of a countermodel from a failed backward proof search in one of our sequent calculi.

In spite of their simplicity, these sequent calculi are well suited for automated theorem proving. A crucial property is that proof search always terminates. To ensure termination, the sequent calculi for $\mathsf{S4}$, $\mathsf{K}+T$ and $\mathsf{KT}+T$ contain a built-in loop-check. Including the loop-check in the calculi has two advantages: It is clearer how and when we have to check for a loop, and it makes it easier to optimise the loop-check for each logic in a perspicuous way.

During backward proof search in these sequent calculi we have to backtrack when a branch fails. We have shown which rules are invertible and when such backtracking is necessary. For an efficient proof search the so-called use-check is crucial. It helps to cut off branches which are caused by unnecessary branchings, i.e. by superfluous backward applications of the ($\wedge$) rules. Further important optimisations are the deletion of duplicated formulas and the extension of the (id) axiom.

In the case of K_t the situation is much more complicated. Backward proof search in the standard tableaux and sequent calculi can lead to sequents with exponential size. Also the termination problem is, in contrast to K, not trivial. We have developed a sequent calculus such that backward proof search terminates and requires only polynomial space. This calculus could certainly be improved in order to make it more suitable for backward proof search.

6

COMPLEXITY

It is a fact that people habitually underestimate the intricacy and complexity that can result from a huge number of interacting units obeying formal rules at very high speeds, relative to our time scale.

D.R. Hofstadter, D.C. Dennett. The mind's I.

6.1 INTRODUCTION

6.1.1 REMARK complexity

In this chapter we discuss the complexities of the validity problems for the logics K, $\mathsf{K}+T$, KT, $\mathsf{S4}$, and K_t (problem complexity). Here the validity problem for L is the problem of deciding whether some formula A in negation normal form is valid. The satisfiability problem for L is analogously the problem of deciding whether a formula A in negation normal form is satisfiable in L. If the complexity of the validity problem for the logic L is C (where C is a complexity class), then there exists no algorithm that 'does better'.

We also investigate upper and lower bounds for the time and space required by proof search algorithms from chapters 4 and 5, in order to show the differences between the underlying calculi (algorithm complexity).

All these complexities are relative to the length of the formula whose provability we want to check.

6.1.2 REMARK problem complexity vs. algorithm complexity

The complexity of a problem is determined by its complexity class. However, for specific algorithms it is sensible to give both bounds for the required time and the required space, since there are algorithms for the same problem that require the same amount of space, but very different amounts of time.

6.1.3 REMARK worst vs. average case complexity

Note that all complexity results in this chapter are worst case complexities.

6.1.4 THEOREM complexity classes

We will use the following complexity classes:

- P (polynomial time)
- NP (non-deterministic polynomial time)
- coNP
- PSPACE (polynomial space)
- EXPTIME (exponential time)

The diagram of these complexity classes is shown below. $\mathsf{A} \longrightarrow \mathsf{B}$ means that A is a subset of B.

$$\begin{tikzcd}
 & \mathsf{EXPTIME} & \\
 & \mathsf{PSPACE} \arrow[u] & \\
\mathsf{NP} \arrow[ur] & & \mathsf{coNP} \arrow[ul] \\
 & \mathsf{P} \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

It is neither known whether any of the subsets in the diagram are proper subsets, nor is it known whether NP and coNP are equal. However, $\mathsf{P} \neq \mathsf{EXPTIME}$. See any book on complexity theory for more information on complexity classes and their interrelationships.

To give bounds for the time and space required by a specific algorithm we use the notions written in brackets behind the complexity classes. For this purpose we also use the notions of exponential space and double exponential time.

6.1.5 THEOREM NP

Assume that for a given problem with size n:

- If there is a solution, then the solution is a member of a set S with an, at most, exponential number (relative to n) of possible solutions.
- We can construct an arbitrary member X out of this set S in polynomial time (relative to n).
- We can check in polynomial time (relative to n) whether this member X is a solution to the problem.

Then the complexity of the problem is in NP (relative to n).

Example

Let A be a formula of CPC with $\text{length}(A) = n$.

Then $\text{card}(\text{vars}(A)) \leq n$. Thus the set S of valuations of these variables contains at most 2^n elements. We can construct any of these valuations in linear time (relative to n). We can check in linear time (relative to n) whether a valuation satisfies the formula A.

Thus complexity of the satisfiability problem for classical propositional logic is in NP.

6.1.6 REMARK coNP and coPSPACE

A problem is in coNP if its complement is in NP.

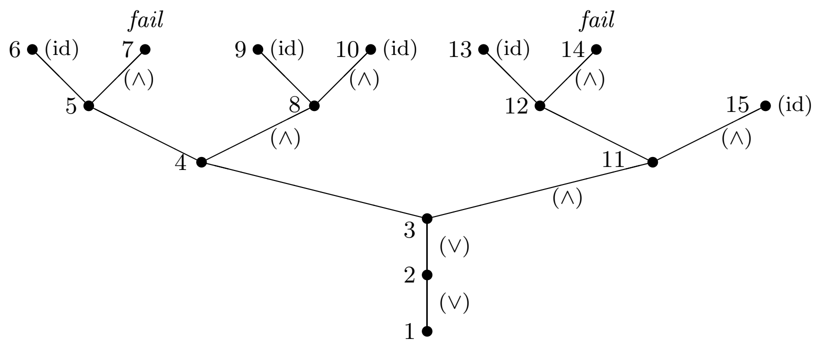
A problem is in coPSPACE if its complement is in PSPACE, and coPSPACE = PSPACE.

Example

If we do backward proof search for the formula A in the usual sequent calculus for CPC, then in each step, one connective is removed from the sequent. The branching factor is 1 or 2 in each node. Thus the search tree has at most 2^n leaves.

We can construct an arbitrary leaf in linear time, and we can check in linear time whether this leaf is an axiom. If it is not an axiom, then A is not provable. Thus the complexity of the validity problem for CPC is in coNP (cp. theorem 6.1.5).

The picture below shows the search tree for the formula $(p_0 \wedge p_1) \vee (p_0 \wedge \neg p_1) \vee (\neg p_0 \wedge p_1)$.



We can construct any of the leaves in linear time by choosing the right branch at each branching. If we choose — by chance — the branch with leaf 7, then we obtain the sequent p_0, p_0, p_1. Since this sequent is not provable the whole formula is not provable.

6.1.7 REMARK invertible rules

We will always assume that we make use of the invertibility of rules of a calculus during backward proof search.

Example

Remark 4.2.7 shows what can happen if we omit this condition.

6.1.8 REMARK restricting (jump) and (jump−)

When discussing backward proof search in graph calculi, then we assume that there are no unnecessary applications of (jump) and (jump−) in the proofs we consider, as we have done in chapter 4.

6.1.9 REMARK use-check

Already in the case of CPC, use-check can lead to an exponential speed-up. The example below is certainly artificial, but practice shows that use-check is crucial in many cases.

In this chapter we will only show some typical modal formula for which use-check proves to be important. We will choose simple examples to explain the basic idea, and therefore one sees at once how the unnecessary branchings could be avoided by simpler means than use-check.

Example

Let A_n be the formula $(p_0 \wedge p_1) \vee \neg p_0 \vee \neg p_1 \vee (p_2 \wedge p_3) \vee (p_4 \wedge p_5) \vee \ldots \vee (p_{2n} \wedge p_{2n+1})$, where $n > 0$. If we apply $(\wedge)$ backwards on $p_0 \wedge p_1$ only if no other conjunction is left, then we obtain a search tree with more than 2^n branches. With use-check we obtain an exponential speed-up.

The reason for the speed-up are the superfluous subformulas. Lemma generation does not help in this example, but the connection method works well since only p_0 and p_1 occur both positively and negatively.

6.1.10 REMARK negation normal form

In this chapter we only consider formulas in negation normal form. Note that for example for the formulas $p_0 \leftrightarrow p_1 \leftrightarrow \ldots \leftrightarrow p_n$ there exists no short negation normal form, i.e. if A is a formula in negation normal form that is equivalent to $p_0 \leftrightarrow p_1 \leftrightarrow \ldots \leftrightarrow p_n$, then the length of A is exponential relative to n.

6.2 K

6.2.1 THEOREM $\mathsf{K}^{\mathcal{G}}$

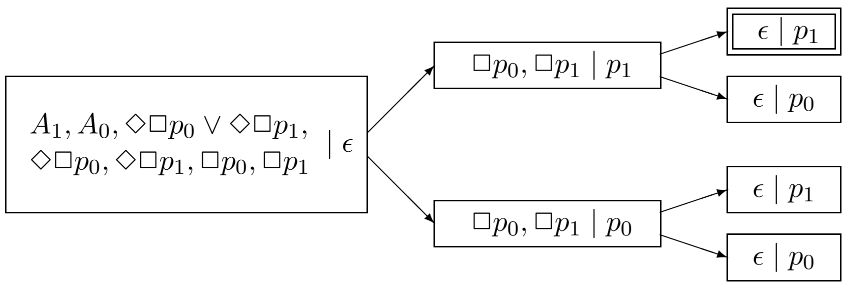
Backward proof search in $\mathsf{K}^{\mathcal{G}}$ (for formulas in negation normal form, with a depth-first strategy, restricting the rules (jump), (jump−), using the invertibility results) can require exponential space and double exponential time.

Proof

We define $A_n :\equiv \bigvee_{i=0,\ldots,n}(\Diamond^i \Box p_0 \vee \Diamond^i \Box p_1)$, Thus we have $A_0 \equiv \Box p_0 \vee \Box p_1$ and $A_{n+1} \equiv A_n \vee (\Diamond^{n+1}\Box p_0 \vee \Diamond^{n+1}\Box p_1)$.

If we do backward proof search in $\mathsf{K}^{\mathcal{G}}$ for the formula A_n for some $n \in \mathbb{N}$, then we finally obtain a K graph with the shape of a complete binary tree with depth n, i.e. with $2^{n+2} - 1$ vertices. Thus exponential space is required for the search. Note that is does not matter in which order we apply the rules backwards (provided (jump), (jump−) are restricted).

In the case of $A_1 \equiv (\Box p_0 \vee \Box p_1) \vee (\Diamond\Box p_0 \vee \Diamond\Box p_1)$ we obtain the K graph below. The actual vertex has been chosen arbitrarily.



$A_1, A_0, \Diamond\Box p_0 \vee \Diamond\Box p_1, \Diamond\Box p_0, \Diamond\Box p_1, \Box p_0, \Box p_1 \mid \epsilon$
$\Box p_0, \Box p_1 \mid p_1$
$\Box p_0, \Box p_1 \mid p_0$
$\epsilon \mid p_1$
$\epsilon \mid p_0$
$\epsilon \mid p_1$
$\epsilon \mid p_0$

If we do backward proof search for $A_n \vee \Diamond^{n+1}(R_1 \wedge R_2)$, where R_1, R_2 are new variables, then we have to apply $(\wedge)$ backwards in each leaf of these K graphs, i.e. the search tree has branches with exponential length, and a double exponential number of nodes. However, backward proof search stops as soon as the first leaf of the search tree is reached since it is not an axiom. In the following we adapt this idea such that a similar search tree with a double exponential number of nodes must be traversed completely.

We define $B_n :\equiv A_n \vee \Diamond Q_1 \vee \bigvee_{i=1,\ldots,n+1}(\Diamond^i(\neg p_0 \wedge \neg Q_i \wedge \Diamond Q_{i+1}))$, where $Q_1, \ldots, Q_{n+1}$ are new variables, i.e. $\mathrm{card}(\{Q_1, \ldots, Q_{n+1}, p_0, p_1\}) = n + 3$.

With this formula we still obtain a complete binary tree as for A_n, but with one 'marked' leaf, namely a leaf that contains a formula that no other leaf contains. To show the idea behind these formulas we show what happens when we do backward proof search for B_1 in $\mathsf{K}^{\mathcal{G}}$. We Have $B_1 \equiv A_1 \vee \Diamond Q_1 \vee \Diamond(\neg p_0 \wedge \neg Q_1 \wedge \Diamond Q_2) \vee \Diamond\Diamond(\neg p_0 \wedge \neg Q_2 \wedge \Diamond Q_3)$. First we apply all rules backwards except $(\wedge)$ and except $(\Diamond)$ with main formula Q_1. The result is the following K graph (the left hand sides are replaced by '$\ldots$'):

$$\begin{tikzcd}
 & & \boxed{\boxed{\epsilon \mid p_1, \neg p_0 \wedge \neg Q_2 \wedge \Diamond Q_3}} \\
 & \boxed{\ldots \mid p_1, \neg p_0 \wedge \neg Q_1 \wedge \Diamond Q_2} \arrow[ur] \arrow[dr] & \\
 & & \boxed{\epsilon \mid p_0, \neg p_0 \wedge \neg Q_2 \wedge \Diamond Q_3} \\
\boxed{\ldots \mid \Diamond Q_1} \arrow[uur] \arrow[ddr] & & \\
 & & \boxed{\epsilon \mid p_1, \neg p_0 \wedge \neg Q_2 \wedge \Diamond Q_3} \\
 & \boxed{\ldots \mid p_0, \neg p_0 \wedge \neg Q_1 \wedge \Diamond Q_2} \arrow[ur] \arrow[dr] & \\
 & & \boxed{\epsilon \mid p_0, \neg p_0 \wedge \neg Q_2 \wedge \Diamond Q_3}
\end{tikzcd}$$

Now we loosen the restriction on $(\wedge)$: We apply it backwards, but only on the branch of the K graph where each vertex (except the root) contains p_0, and we still apply (id) and $(\Diamond)$ backwards whenever possible. Only one branch of the corresponding search tree does not end immediately in an axiom, but in the K graph below. We show the $(\Diamond)$ formulas that are put on the left hand sides during these steps.

$$\begin{tikzcd}
 & & \boxed{\epsilon \mid p_1, \neg p_0 \wedge \neg Q_2 \wedge \Diamond Q_3} \\
 & \boxed{\ldots \mid p_1, Q_1, \neg p_0 \wedge \neg Q_1 \wedge \Diamond Q_2} \arrow[ur] \arrow[dr] & \\
 & & \boxed{\epsilon \mid p_0, \neg p_0 \wedge \neg Q_2 \wedge \Diamond Q_3} \\
\boxed{\Diamond Q_1, \ldots \mid \epsilon} \arrow[uur] \arrow[ddr] & & \\
 & & \boxed{\epsilon \mid p_1, Q_2, \neg p_0 \wedge \neg Q_2 \wedge \Diamond Q_3} \\
 & \boxed{\Diamond Q_2, \ldots \mid p_0, Q_1} \arrow[ur] \arrow[dr] & \\
 & & \boxed{\boxed{\Diamond Q_3, \ldots \mid p_0, Q_2}}
\end{tikzcd}$$

Now we construct the complete search tree for this K graph. We always apply (id) immediately. There exist branches in the search tree where $(\Diamond)$ is not applied backwards and that does not end in (id). One of these branches ends in

$$\begin{tikzcd}
 & & \boxed{\ldots \mid p_1, \neg p_0} \\
 & \boxed{\ldots \mid p_1, Q_1, \neg p_0} \arrow[ur] \arrow[dr] & \\
 & & \boxed{\ldots \mid p_0, \neg Q_2} \\
\boxed{\Diamond Q_1, \ldots \mid \epsilon} \arrow[uur] \arrow[ddr] & & \\
 & & \boxed{\ldots \mid p_1, Q_2, \neg p_0} \\
 & \boxed{\Diamond Q_2, \ldots \mid p_0, Q_1} \arrow[ur] \arrow[dr] & \\
 & & \boxed{\boxed{\Diamond Q_3, \ldots \mid p_0, Q_2}}
\end{tikzcd}$$

Now, exactly one world contains $\Diamond Q_3$, i.e. we have succeeded in marking one of the worlds. If we do backward proof search for $C_n \equiv B_n \vee \Diamond^{n+1}\Box\neg Q_{n+1} \vee \Diamond^{n+2}(R_1 \wedge R_2)$, where R_1, R_2 are new variables, then we obtain for C_1 the K graph

$$\begin{tikzcd}
 & & \boxed{\ldots \mid p_1, \neg p_0, \Box\neg Q_3, \Diamond(R_1 \wedge R_2)} \\
 & \boxed{\ldots \mid p_1, Q_1, \neg p_0} \arrow[ur] \arrow[r] & \boxed{\ldots \mid p_0, \neg Q_2, \Box\neg Q_3, \Diamond(R_1 \wedge R_2)} \\
\boxed{\Diamond Q_1, \ldots \mid \epsilon} \arrow[ur] \arrow[dr] & & \\
 & \boxed{\Diamond Q_2, \ldots \mid p_0, Q_1} \arrow[r] \arrow[dr] & \boxed{\ldots \mid p_1, Q_2, \neg p_0, \Box\neg Q_3, \Diamond(R_1 \wedge R_2)} \\
 & & \boxed{\boxed{\Diamond Q_3, \ldots \mid p_0, Q_2, \Box\neg Q_3, \Diamond(R_1 \wedge R_2)}}
\end{tikzcd}$$

on this branch of the search tree. Now we can apply $(\wedge)$ backwards in each world with main formula $R_1 \wedge R_2$. Only at the end do we apply $(\Box)$ with main formula $\Box Q_{n+1}$. Thus the search tree for C_n has more than $2^{2^{n+1}}$ leaves, and we traverse the whole search tree. Therefore double exponential time is required for backward proof search.

Use-check would help to solve the problem, since the $(\wedge)$ applications with main formula $R_1 \wedge R_2$ are in fact completely independent.

6.2.2 THEOREM $\mathsf{K}^{\mathcal{S},2}$

Backward proof search in $\mathsf{K}^{\mathcal{S},2}$ (for formulas in negation normal form, with a depth-first strategy, using the invertibility results) requires at most polynomial space.

Proof

We use the function m from the proof of termination of backward proof search to show that the length of the branches of the search tree is at most length(A).

The branching degree is at most length(A) in all nodes of the search tree: It is 1 if we apply $(\vee)$ or $(\Diamond)$ backwards, 2 if we apply $(\wedge)$ backwards, and at most length(A) if we apply $(\Box)$ backwards.

Thus the search tree has at most depth length(A), at most branching degree length(A), and for each sequent Γ that can occur we have length$(\Gamma) \leq$ length(A), i.e. polynomial space is sufficient for backward proof search in $\mathsf{K}^{\mathcal{S},2}$.

6.2.3 REMARK $\mathsf{K}^{\mathcal{G}}$ vs. $\mathsf{K}^{\mathcal{S},2}$

The formulas C_n from the proof of 6.2.1 cause no problems in the case of backward proof search in $\mathsf{K}^{\mathcal{S},2}$, since the backward applications of $(\wedge)$ with main formula $R_1 \wedge R_2$ on different branches are done independently. This effect is quite frequent and must be taken into account when implementing a decision procedure based on labelled tableaux.

6.2.4 THEOREM validity problem

The complexity of the validity problem for K is PSPACE.

Proof

This result was proved in [Lad77]. The upper limit follows immediately from theorem 6.2.2. See [Hud96] for tighter bounds.

6.2.5 THEOREM size of countermodel

If a formula A is satisfiable K, then:

- There exists a K model $\mathcal{M}$ with $\mathsf{K}, \mathcal{M} \models A$ and diam$(\mathcal{M}) \leq$ length(A).
- There exists a K model $\mathcal{M}$ with card$(\mathcal{M}) \leq 2^{\text{length}(A)}$.

Proof

Backward proof search in $\mathsf{K}^{\mathcal{S},2}$ for A gives us a model $\mathcal{M}$ with $\text{diam}(\mathcal{M}) \leq \text{length}(A)$.

See [HM92] for a proof of the second statement. Note that the countermodels we obtain from backward proof search with $\mathsf{K}^{\mathcal{S},2}$ can be larger. For example, take a variant of the formulas A_n from the proof of theorem 6.2.1 where each world has three successors.

6.3 $\mathsf{K} + T$

6.3.1 THEOREM size of countermodels

For all $e \in \mathbb{N}$ there exists a formula A and a theory T with $\mathsf{K} + T \not\models A$ such that:

$$(\mathsf{K}, \mathcal{M} \models T \text{ and } \mathsf{K}, \mathcal{M} \not\models A) \quad \Rightarrow \quad \text{diam}(\mathcal{M}) \geq (\text{length}(T) + \text{length}(A))^e$$

Proof

This theorem is an immediate consequence of the proof of theorem 7.2.9 in the following chapter.

6.3.2 REMARK validity problem

The complexity of the validity problem for the logic $\mathsf{K} + T$ is $\mathsf{EXPTIME}$ (personal communication F. Baader).

6.3.3 REMARK use-check

Use-check is especially important if the theory contains disjunctions. They cause a branching after every backward application of the $(\Box)$ rule.

Example

Let A be the formula $\Box^n(p_0 \vee p_1 \vee p_2)$ and T the theory $p_0 \vee p_1$.

If we do backward proof search for A in $(\mathsf{K} + T)^{\mathcal{S},2}$, then we obtain a search tree with 2^{n+1} branches. The branchings are caused by the theory: At the beginning and after every of the n backward applications of $(\Box)$ we add the formula $\neg p_0 \wedge \neg p_1$, and thus have to apply $(\wedge)$ backwards. With use-check we consider only two of these branches and cut off all the others, i.e. we have an exponential speed-up.

Note that use-check is applicable although there are no superfluous subformulas involved. This is also an example where lemma generation does not help at all.

6.4 KT

6.4.1 THEOREM $\mathsf{KT}^{\mathcal{G}}$

Backward proof search in $\mathsf{KT}^{\mathcal{G}}$ (for formulas in negation normal form, with a depth-first strategy, restricting the rules (jump), (jump−) and using the invertibility results) can require exponential space and double exponential time.

Proof

We only sketch the proof. The idea is the same as in the proof of theorem 6.2.1. In contrast to K, a $\Diamond A$ can now also mean the world itself and not only its successor. We put new variables as marks into all worlds so that all worlds with the same distance from the root have the same mark. If $\Diamond A$ is a formula that will occur in a formula with mark Q, then we replace it by $\Diamond(A \wedge \neg Q)$ in order to make sure that there is a proper successor with A.

6.4.2 THEOREM $\mathsf{KT}^{\mathcal{S},2}$

Backward proof search in $\mathsf{KT}^{\mathcal{S},2}$ requires at most polynomial space and exponential time.

Proof

Analogous to the proof of theorem 6.2.2, using the measure m from the proof of theorem 5.4.16.

6.4.3 THEOREM validity problem

The complexity of the validity problem for KT is PSPACE.

Proof

This result was proved in [Lad77]. The upper limit follows immediately from theorem 6.4.2. See [Hud96] for tighter bounds.

6.4.4 THEOREM duplicate $\Diamond$ formulas

Note that it is crucial for theorem 6.4.2 to avoid duplicate $\Diamond$ formulas on the left hand side of the sequents in the calculus $\mathsf{KT}^{\mathcal{S},2}$. The sequents we obtain during backward proof search for a sequent of the form $\epsilon \mid A$ in $\mathsf{KT}^{\mathcal{S}}$ can have exponential length (relative to length(A)).

Proof

We do backward proof search for $\epsilon \mid \Diamond^{2n} p_0 \vee \Box^n p_1$ in $\mathsf{KT}^{\mathcal{S}}$. The search tree consists of n backward applications of $(\Box)$, with backward applications of $(\Diamond)$ in between. After $n_\Box$ backward applications of $(\Box)$ and having applied $(\Diamond)$ backwards as often as possible, we have a sequent with $\binom{2n}{n_\Box}$ copies of the variable p_0. If $n_\Box = n$, we thus have $\binom{2n}{n} = \frac{(2n)!}{n! \cdot n!} > 2^n$ copies of p_0. This 'explosion' is related to the problems with the mapping of KT in K (see theorem 7.3.5).

See figure 6.a for the non-proof of $\epsilon \mid \Diamond^6 p_0 \vee \Box^3 p_1$ in $\mathsf{KT}^{\mathcal{S}}$.

6.4.5 THEOREM size of countermodel

If a formula A is satisfiable KT, then:

- There exists a KT model $\mathcal{M}$ with $\mathsf{KT}, \mathcal{M} \models A$ and polynomial diam$(\mathcal{M})$.
- There exists a KT model $\mathcal{M}$ with card$(\mathcal{M}) \leq 2^{\text{length}(A)}$.

Proof

The first part follows from the proof of theorem 5.4.16. See [HM92] for a proof of the second statement.

6.4.6 REMARK use-check

Conjunctions inside nested $\Diamond$ can cause many unnecessary branchings, although the formula needs not be superfluous.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Diamond^3 p_0, \Diamond^2 p_0, \Diamond p_0, \Diamond^2 p_0, \Diamond p_0, \Diamond p_0, \Diamond^2 p_0, \Diamond p_0, \Diamond p_0, \Diamond p_0, \Diamond^2 p_0, \ldots \mid p_0, p_0, p_0, p_0, p_0, p_0, p_0, p_0, p_0, p_0, p_0, p_0, p_0, p_0, p_0, p_0, p_0, p_0, p_0, p_0, p_1}{\vdots}\;(\Diamond)}{\epsilon \mid \Diamond^3 p_0, \Diamond^2 p_0, \Diamond p_0, p_0, \Diamond^2 p_0, \Diamond p_0, p_0, \Diamond p_0, p_0, p_0, \Diamond^2 p_0, \Diamond p_0, p_0, \Diamond p_0, p_0, p_0, \Diamond p_0, p_0, p_0, p_0, p_1}\;(\Diamond)}{\Diamond^4 p_0, \Diamond^3 p_0, \Diamond^2 p_0, \Diamond p_0, \Diamond^3 p_0, \Diamond^2 p_0, \Diamond p_0, \Diamond^2 p_0, \Diamond p_0, \Diamond p_0, \Diamond^3 p_0, \Diamond^2 p_0, \Diamond p_0, \ldots \mid p_0, p_0, p_0, p_0, p_0, p_0, p_0, p_0, p_0, p_0, p_0, p_0, p_0, p_0, p_0, \Box p_1}\;(\Box)}{\vdots}\;(\Diamond)}{\epsilon \mid \Diamond^4 p_0, \Diamond^3 p_0, \Diamond^2 p_0, \Diamond p_0, p_0, \Diamond^3 p_0, \Diamond^2 p_0, \Diamond p_0, p_0, \Diamond^2 p_0, \Diamond p_0, p_0, \Diamond p_0, p_0, p_0, \Box p_1}\;(\Diamond)}{\Diamond^5 p_0, \Diamond^4 p_0, \Diamond^3 p_0, \Diamond^2 p_0, \Diamond p_0, \Diamond^4 p_0, \Diamond^3 p_0, \Diamond^2 p_0, \Diamond p_0, \Diamond^3 p_0, \Diamond^2 p_0, \Diamond p_0, \Diamond^2 p_0, \Diamond p_0, \Diamond p_0 \mid p_0, p_0, p_0, p_0, p_0, p_0, \Box^2 p_1}\;(\Box)}{\vdots}\;(\Diamond)}{\epsilon \mid \Diamond^5 p_0, \Diamond^4 p_0, \Diamond^3 p_0, \Diamond^2 p_0, \Diamond p_0, p_0, \Box^2 p_1}\;(\Diamond)}{\Diamond^6 p_0, \Diamond^5 p_0, \Diamond^4 p_0, \Diamond^3 p_0, \Diamond^2 p_0, \Diamond p_0 \mid p_0, \Box^3 p_1}\;(\Box)}{\vdots}\;(\Diamond)}{\epsilon \mid \Diamond^6 p_0, \Box^3 p_1}\;(\Diamond)$$

Figure 6.a: The non-proof in $\mathsf{KT}^{\mathcal{S}}$ of the example in theorem 6.4.4.

Example

Let A be the formula $\Box^n(p_0 \vee p_1), \Diamond^n(\neg p_0 \wedge \neg p_1)$.

If we do backward proof search for A in $\mathsf{KT}^{\mathcal{S},2}$, then we obtain a search tree with 2^{n+1} branches. At the beginning and after every of the n backward applications of $(\Box)$ we unpack $\neg p_0 \wedge \neg p_1$ by applying $(\Diamond)$ backwards. With use-check we obtain only two branches, i.e. we have an exponential speed-up.

Note that the subformula $\neg p_0 \wedge \neg p_1$ is not superfluous. The branching is needed in the proof, but only once and not $n+1$ times.

6.5 S4

6.5.1 REMARK graph calculi

Backward proof search in $\mathsf{S4}^{\mathcal{G}}$ does in general not terminate. If we introduced some sort of loop-check, then we would obtain similar results as for $\mathsf{K}^{\mathcal{G}}$.

6.5.2 THEOREM $\mathsf{S4}^{\mathcal{S},3}$

Backward proof search in $\mathsf{S4}^{\mathcal{S},3}$ requires at most polynomial space.

Proof

Analogous to the proof of theorem 6.2.2, using the measure m from the proof of theorem 5.6.17.

6.5.3 THEOREM validity problem

The complexity of the validity problem for S4 is PSPACE.

Proof

This result was proved in [Lad77]. The upper limit follows immediately from 6.5.2. See [Hud96] for tighter bounds.

6.5.4 THEOREM size of countermodel

If a formula A is satisfiable S4, then:

- There exists a S4 model $\mathcal{M}$ with $\mathsf{S4}, \mathcal{M} \models A$ and polynomial diam($\mathcal{M}$).
- There exists a S4 model $\mathcal{M}$ with card($\mathcal{M}$) $\leq 2^{\text{length}(A)}$.

Proof

The first part follows from the proof of theorem 5.6.17. See [HM92] for a proof of the second statement.

6.5.5 THEOREM $(\Box)$ of $\mathsf{S4}^{\mathcal{S},3}$

Let $\mathsf{S4}^{\mathcal{S},2+}$ be the calculus $\mathsf{S4}^{\mathcal{S},3}$ without the multiset $\Box\Pi$ in the premise of the $(\Box)$ rule. Then backward proof search in $\mathsf{S4}^{\mathcal{S},3}$ can be exponentially faster than backward proof search in $\mathsf{S4}^{\mathcal{S},2+}$.

Proof

Let A be the formula $\Box(\Diamond\Box p_0 \vee \ldots \vee \Diamond\Box p_n)$ and B the formula $\Box(p_0 \vee \neg p_0)$.

If we do backward proof search for $A \vee B$ in $\mathsf{S4}^{\mathcal{S},3}$, then we first have to apply $(\vee)$ backwards and obtain $\epsilon \mid \epsilon \mid A, B$. Now we apply $(\Box)$ backwards with main formula $\Box(\Diamond\Box p_0 \vee \ldots \vee \Diamond\Box p_n)$. We obtain the sequent $A \mid \epsilon \mid \Diamond\Box p_0 \vee \ldots \vee \Diamond\Box p_n$. The search tree for this sequent has n leaves. Such a leaf has the form $\Box p_0, \ldots, \Box p_n \mid \Diamond\Box p_0, \ldots, \Diamond\Box p_n \mid \Box p_0, \ldots, \Box p_n$. The search stops in these leaves, since the history already contains all the $\Box$ formulas.

If we do backward proof search in $\mathsf{S4}^{\mathcal{S},2+}$ instead, then the corresponding sequents are $\Box p_i \mid \Diamond\Box p_0, \ldots, \Diamond\Box p_n \mid \Box p_0, \ldots, \Box p_n$, i.e. there are still $n-1$ possible ways to apply $(\Box)$ backwards.

After backtracking through this search tree, we apply $(\Box)$ backwards on $\Box(p_0 \vee \neg p_0)$ and obtain an axiom. The search tree for $\mathsf{S4}^{\mathcal{S},3}$ has $n+1$ leaves, the one for $\mathsf{S4}^{\mathcal{S},2+}$ has about $n!$ leaves.

6.5.6 REMARK use-check

Conjunctions inside a $\Diamond$ can cause many unnecessary branchings, although the formula needs not be superfluous.

Example

Let A be the formula $\Box^n(p_0 \vee p_1), \Diamond(\neg p_0 \wedge \neg p_1)$.

If we do backward proof search for A in $\mathsf{S4}^{\mathcal{S},3}$, then we obtain a search tree with 2^{n+1} branches. At the beginning and after every of the n backward application of $(\Box)$ we unpack $\neg p_0 \wedge \neg p_1$ by applying $(\Diamond)$ backwards. With use-check we obtain only two branches, i.e. we have an exponential speed-up.

Of course $\Box^n A$ is equivalent to $\Box A$ in $\mathsf{S4}$. In order to make the example less obvious we can for example replace each subformula of the form $\Box A$ by $\Box A \vee Q$, where Q is neither p_0 nor p_1.

6.6 K_t

6.6.1 THEOREM graph calculi

Backward proof search in $\mathsf{K}_t^{\mathcal{G}}$, (for formulas in negation normal form, with a depth-first strategy, restricting the rules (jump), (jump$-$), restricting the rules $(\Box)$, $(\blacksquare)$ (see 4.7.3), using the invertibility results) can require exponential space and double exponential time.

Proof

We can reuse the proof of theorem 6.2.1.

6.6.2 THEOREM $\mathsf{K}_t^{\mathcal{S}}$

Backward proof search in $\mathsf{K}_t^{\mathcal{S}}$ requires at most polynomial space.

Proof

Assume that we do backward proof search for $\boxed{\boxed{\epsilon \mid \epsilon \mid A^0}}$ in $\mathsf{K}_t^{\mathcal{S}}$. From the proof of theorem 5.7.22 follows that the branches of the search tree have polynomial length (relative to length(A)). Since the branching degree is at most length(A) and each sequent has polynomial size our assertion follows immediately. For this proof it is important that there is only a polynomial number of (guess) applications between two $(\Box)$ applications.

6.6.3 THEOREM validity problem

The complexity of the validity problem for K_t is in PSPACE.

Proof

The lower limit follows from the corresponding result about K (see 6.2.4, [Lad77]). The upper limit was proved in [Spa93]; it follows also immediately from 6.6.2.

6.7 OTHER LOGICS

6.7.1 THEOREM other logics

The complexity of the validity problem for other logics, together with pointers to the literature:

logic	complexity of the validity problem	literature
CPC	coNP	
S5	coNP	[Lad77]
K_n	PSPACE	[HM92]
KT_n	PSPACE	[HM92]
$\mathsf{S4}_n$	PSPACE	[HM92]
$\mathsf{S5}_n$	PSPACE	[HM92]
PLTL	PSPACE	[SC85]
CTL	EXPTIME	[Eme90]
IPC	PSPACE	[Sta79], [Hud93]
linear logic	undecidable	[Tro92]

In [FHMV96] and [HM92] the complexity of variants of the multimodal logics is investigated. Some results concerning the relation between number of variables, modal depth and complexity can be found in [Hal95].

6.8 SUMMARY

It is known that the complexity of the validity problem of K, KT, S4, and K_t is PSPACE (see [Lad77], [Spa93]).

Backward proof search in our sequent calculi $\mathsf{K}^{\mathcal{S},2}$, $\mathsf{KT}^{\mathcal{S},2}$, $\mathsf{S4}^{\mathcal{S},3}$, $\mathsf{K}_t^{\mathcal{S}}$ requires at most polynomial space. The space required for backward proof search in our graph calculi can be much higher.

This is not by chance: All rules of the graph calculi $\mathsf{K}^{\mathcal{G}}$, $\mathsf{KT}^{\mathcal{G}}$, $\mathsf{S4}^{\mathcal{G}}$, $\mathsf{K}_t^{\mathcal{G}}$ are invertible. Since the branching degree of the search tree is small, the length of the branches of the search tree must sometimes be exponential, unless coNP = PSPACE. The branches of the search tree in the case of the sequent calculi, on the other hand, have polynomial length. This is possible because all these calculi contain non-invertible rules.

7

EMBEDDINGS

em·bed (imbéd) *pres. part.* **em·bed·ding** *past and past part.* **em·bed·ded** *v.t.* to fix (something) firmly into a surrounding substance.

Webster Encyclopedic Dictionary.

7.1 INTRODUCTION

7.1.1 REMARK embeddings and decision procedures

Assume that we want to know whether a formula A is valid in the logic L_1. If f is an embedding of L_1 in L_2, then A is valid in L_1 iff $f(A)$ is valid in L_2. Thus we can decide the validity of A by applying a decision procedure for L_2 on $f(A)$.

In practice, it is of course important that computing $f(A)$ is not too time-consuming. Moreover, we have to take into account that it cannot be easier to decide the validity of formulas of L_2 than the one of formulas of L_1 (provided that the complexity of f is reasonably restricted). From the point of view of efficiency, it is therefore not advisable to make use of embeddings. But there are also advantages: If there already exists a theorem prover for L_2 that was tested thoroughly, using an embedding of L_1 in L_2 is an easy and reliable method. And if the existing theorem prover for L_2 deals efficiently with the required subset, the resulting procedure will be efficient, too.

7.1.2 DEFINITION embedding, mapping

Assume that L_1, L_2 are two logics without theories, and $L_3, L_4 \in \{\mathsf{K}, \mathsf{KT}, \mathsf{S4}\}$.

- $f : \mathrm{Fml}_{L_1} \to \mathrm{Fml}_{L_2}$ is an embedding of L_1 in L_2
  iff $\forall A \in \mathrm{Fml}_{L_1} : (*)$ and $(L_1 \models A \Leftrightarrow L_2 \models f(A))$.
- $f : \mathrm{Fml}_{L_3} \times \mathrm{Th}_{L_3} \to \mathrm{Fml}_{L_1}$ is an embedding of L_3 plus theories in L_1
  iff $\forall A \in \mathrm{Fml}_{L_3} : \forall T \in \mathrm{Th}_{L_3} : (*)$ and $(L_3 + T \models A \Leftrightarrow L_1 \models f(A, T))$.
- $f : \mathrm{Fml}_{L_1} \to \mathrm{Fml}_{L_3} \times \mathrm{Th}_{L_3}$ is an embedding of L_1 in L_3 plus theories
  iff $\forall A \in \mathrm{Fml}_{L_3} : (*)$ and $(L_1^{\mathcal{H}} \models A \Leftrightarrow L_3 + \mathrm{el}_2(f(A)) \models \mathrm{el}_1(f(A)))$.

- $f : \mathrm{Fml}_{L_3} \times \mathrm{Th}_{L_3} \to \mathrm{Fml}_{L_4} \times \mathrm{Th}_{L_4}$ is an embedding of L_3 plus theories in L_4 plus theories iff $\forall A \in \mathrm{Fml}_{L_3} : \forall T \in \mathrm{Th}_{L_3} : (*)$ and $(L_3 + T \models A \Leftrightarrow L_4 + \mathrm{el}_2(f(A,T)) \models \mathrm{el}_1(f(A,T)))$.

where $(*)$ means that $f(A)$ and $f(A,T)$ can be computed in polynomial time relative to length(A) and length(nnf(A)) + length(T), respectively.

If the condition $(*)$ about the complexity of $f(A)$ or $f(A,T)$ is not satisfied, then we will speak of a mapping instead of an embedding.

7.1.3 REMARK complexity condition

The condition $(*)$ about the complexity of $f(A)$ and $f(A,T)$ in definition 7.1.2 is important. Note that there exists a mapping of L_1 in L_2 for all the logics L_1, L_2 we consider. We simply define

$$f(A) :\equiv \begin{cases} \mathsf{true} & L_1 \models A \\ \mathsf{false} & L_1 \not\models A \end{cases}$$

These are of course not the functions we are interested in.

7.1.4 THEOREM non-existence of embeddings

Assume that L_1, L_2 are two logics such that:

- The complexity of the validity problem for L_1 is PSPACE.
- The complexity of the validity problem for L_1 is coNP.

Then there exists no embedding of L_1 in L_2, unless $\mathsf{coNP} = \mathsf{PSPACE}$ (cp. chapter 6).

Example

There exists no embedding of K in CPC.

7.2 $\mathsf{K} + T$

7.2.1 THEOREM mapping of K plus theories in K

Assume that A, $B_1, \dots, B_n$ are formulas in negation normal form and that T is the multiset $B_1, \dots, B_m$. We set $C :\equiv B_1 \wedge \dots \wedge B_m$ and $n(A,T) :=$ (length(A) + length(nnf$(\neg T)$)) $\cdot$ $2^{\mathrm{length}(A)+\mathrm{length}(\mathrm{nnf}(\neg T))}$. Then we have:

$$\mathsf{K} + T \models A \quad \Leftrightarrow \quad \mathsf{K} \models C \wedge \Box C \wedge \dots \wedge \Box^{n(A,T)} C \to A$$

Proof

We prove

$$(\mathsf{K} + B_1, \dots, B_m)^{\mathcal{S},2} \vdash \epsilon \mid A \;\Leftrightarrow\; \mathsf{K}^{\mathcal{S},2} \vdash D, \Diamond D, \dots, \Diamond^{n(A,T)} D, A$$

where $D \equiv \mathrm{nnf}(\neg C)$. The theorem follows then immediately with the equivalences from chapter 4 and chapter 5.

'$\Rightarrow$':

Since $(\mathsf{K} + B_1, \dots, B_n)^{\mathcal{S},2} \vdash \epsilon \mid A$, we find a proof when doing backward proof search for $\epsilon \mid A$ in $(\mathsf{K} + B_1, \dots, B_n)^{\mathcal{S},2}$. For each branch in this proof we count the number of $(\Box)$ applications on the branch. Let m be the maximum of these numbers.

Now compare backward proof search in $(\mathsf{K}+B_1,\ldots,B_n)^{\mathcal{S},2}$ for $\epsilon \mid A$ and backward proof search in $\mathsf{K}^{\mathcal{S},2}$ for $\epsilon \mid D, \Diamond D, \ldots, \Diamond^{n(A,T)}D, A$. The only difference is that in the former we have $\mathrm{nnf}(\neg T)$ in the premise of each $(\Box)$ application, and in the latter we have a D (and some $\Diamond^i D$) in each premise of a $(\Box)$ application. Since $\mathrm{nnf}(\neg T)$ is essentially D, we obtain $\mathsf{K}^{\mathcal{S},2} \vdash \epsilon \mid D, \Diamond D, \ldots, \Diamond^m D, A$. The condition in the $(\Box)$ rule of $(\mathsf{K}+B_1,\ldots,B_n)^{\mathcal{S},2}$ makes sure that $m \leq n(A,T)$. Therefore $\mathsf{K}^{\mathcal{S},2} \vdash \epsilon \mid D, \Diamond D, \ldots, \Diamond^{n(A,T)}D, A$.

'$\Leftarrow$':

We can use the same idea as in the first part of the proof, but we do not have to worry whether $n(A,T)$ is large enough.

7.2.2 REMARK non-existence of standard embedding

In the following we show that we cannot replace the function n in theorem 7.2.1 by a function with values that are polynomial in $\mathrm{length}(A) + \mathrm{length}(\mathrm{nnf}(\neg T))$.

We will construct a theory $T_{\langle b_1,\ldots,b_m\rangle}$ that depends on natural numbers $b_1,\ldots,b_m$. Every model which shows that $\mathsf{K}+T_{\langle b_1,\ldots,b_m\rangle} \not\models P$ contains a chain of $b_1\cdot\ldots\cdot b_m$ worlds (if $b_1,\ldots,b_m$ are pairwise prime). Then we show that it is possible to choose $b_1,\ldots,b_m$ such that their product is exponential relative to the length of $T_{\langle b_1,\ldots,b_m\rangle}$.

7.2.3 DEFINITION $T_{\langle b_1,\ldots,b_m\rangle}$

If $m, b_1,\ldots,b_m \in \{2,3,\ldots\}$ and $P, Q_1,\ldots,Q_m, R$ are different variables, then we define $A_{i,j} :\equiv \Box^j Q_i \wedge \bigwedge_{k=1}^{j-1}(\Box^k \neg Q_i)$ for all $i \in \{1,\ldots,n\}$ and $j \in \mathbb{N}$.

Then:

$$T_{\langle b_1,\ldots,b_m\rangle} := \Diamond R, \neg Q_1 \vee A_{1,b_1}, \ldots, \neg Q_m \vee A_{m,b_m}, P \vee (A_{1,b_1} \wedge \ldots \wedge A_{m,b_m})$$

7.2.4 LEMMA $\mathsf{K}+T_{\langle b_1,\ldots,b_m\rangle} \not\models P$

If $m, b_1,\ldots,b_m \in \{2,3,\ldots\}$, then $\mathsf{K}+T_{\langle b_1,\ldots,b_m\rangle} \not\models P$.

Proof

It is always possible to construct a countermodel of $T_{\langle b_1,\ldots,b_m\rangle}$ and P analogous to the one in the example below.

Example

Assume that $m = 2$, $b_1 = 3$, $b_2 = 5$.

Then $T_{\langle 3,5\rangle} = \Diamond R, \neg Q_1 \vee A_{1,b_1}, \neg Q_2 \vee A_{2,b_2}, P \vee A_{1,b_1} \wedge A_{2,b_2}$,
where $A_{1,b_1} \equiv \Box^3 Q_1 \wedge (\Box\neg Q_1 \wedge \Box^2 \neg Q_1)$ and $A_{1,b_1} \equiv \Box^5 Q_2 \wedge (\Box\neg Q_2 \wedge \Box^2\neg Q_2 \wedge \Box^3\neg Q_2 \wedge \Box^4 \neg Q_2)$.

Then $\langle\{w_0,\ldots,w_{15}\}, \mathcal{R}, v\rangle$, where $w_0\mathcal{R}w_1$, $w_1\mathcal{R}w_2$, $\ldots$, $w_{14}\mathcal{R}w_{15}$, $w_{15}\mathcal{R}w_1$, and where the valuation v is given by the following table, is a typical countermodel of $T_{\langle 3,5\rangle}$ and P. (A '?' in the table means that the value does not matter.)

I	w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}	w_{12}	w_{13}	w_{14}	w_{15}
P	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Q_1	?	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1
Q_2	?	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1
R	?	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Because of the formula $\Diamond R$ there must always exist an accessible world. In a countermodel of $T_{\langle 3,5\rangle}$ and P there must be a world where P is false, called w_0 in this example. The formula

$P \lor A_{1,3} \land A_{2,5}$ at the beginning, and afterwards, the formulas $\neg Q_1 \lor A_{1,3}$ and $\neg Q_2 \lor A_{2,5}$ effect that Q_1 is exactly true in every third world and that Q_2 is exactly true in every fifth world, respectively. This leads to a loop of $3 \cdot 5$ different worlds. Note that we only get $b_1 \cdot \ldots \cdot b_m$ worlds if the numbers $b_1, \ldots, b_m$ are pairwise prime.

7.2.5 DEFINITION $G(b_1, \ldots, b_m)$

If $m, b_1, \ldots, b_m \in \{2, 3, \ldots\}$, then $G(b_1, \ldots, b_m)$ stands for

$$\forall i, j \in \{1, \ldots, m\} : (i < j \to \gcd(b_i, b_j) = 1)$$

i.e. the natural numbers $b_1, \ldots, b_m$ are pairwise prime.

7.2.6 LEMMA size of countermodels

If $m, b_1, \ldots, b_m \in \{2, 3, \ldots\}$ and $G(b_1, \ldots, b_m)$, then:

$$\mathsf{K}, \mathcal{M} \models T_{\langle b_1, \ldots, b_m \rangle} \text{ and } \mathsf{K}, \mathcal{M} \not\models P \quad \Rightarrow \quad \operatorname{diam}(\mathcal{M}) \geq \prod_{i=1}^{m}(b_i)$$

Proof

Assume that $\mathsf{K}, \mathcal{M} \models T_{\langle b_1, \ldots, b_m \rangle}$ $(*)$ and that $\mathsf{K}, \mathcal{M} \not\models P$.

(1) $\mathsf{K}, \mathcal{M} \models T_{\langle b_1, \ldots, b_m \rangle}$
$\Rightarrow\ \forall w \in W : w \models \Diamond R$
$\Rightarrow\ \forall w \in W : \exists w' \in W : w \mathcal{R} w'$.

(2) $\mathsf{K}, \mathcal{M} \not\models P$
$\Rightarrow\ \exists w_0 \in W : w_0 \models \neg P$
$\overset{(*)}{\Rightarrow}\ \exists w_0 \in W : w_0 \models \neg P \land (P \lor A_{1,b_1} \land \ldots \land A_{m,b_m})$
$\Rightarrow\ \exists w_0 \in W : w_0 \models A_{1,b_1} \land \ldots \land A_{m,b_m}$

(3) Assume $w, w' \in W$ and $d_1 \in \{1, \ldots, b_1\}$, $\ldots$, $d_m \in \{1, \ldots, b_m\}$ such that $w \mathcal{R} w'$ and $w \models A_{1,d_1} \land \ldots \land A_{m,d_m}$. Then we have for all $i \in \{1, \ldots, m\}$:

- $d_i > 1$: $w \models A_{i,d_i} \Rightarrow w \models \Box^{d_i} Q_i \land \bigwedge_{k=1}^{d_i-1}(\Box^k \neg Q_i) \Rightarrow w' \models \Box^{d_i-1} Q_i \land \bigwedge_{k=1}^{d_i-2}(\Box^k \neg Q_i) \Rightarrow w' \models A_{i,d_i-1}$.
- $d_i = 1$: $w \models A_{i,d_i} \Rightarrow w \models \Box Q_i \Rightarrow w' \models Q_i \overset{(*)}{\Rightarrow} w' \models Q_i \land (\neg Q_i \lor A_{i,b_i}) \Rightarrow w' \models A_{i,b_i}$.

(4) Assume $w \in W$, $i \in \{1, \ldots, m\}$ and $d_i, d'_i \in \{1, \ldots, b_i\}$, $d_i \leq d'_i$ such that $w \models A_{i,d_i}$ and $w \models A_{i,d'_i}$.

If $d_i < d'_i$, then $w \models \Box^{d_i} Q_i$ and $w \models \Box^{d_i} \neg Q_i$. With (1) follows $\exists w' \in W : (w' \models Q_i$ and $w' \models \neg Q_i)$. Contradiction. Thus $d_i = d'_i$.

We define $c := \prod_{i=1}^{m}(b_i)$. Because of (1), (2) there are $w_0, w_1, \ldots, w_{c-1} \in W$ such that $w_0 \mathcal{R} w_1$, $w_1 \mathcal{R} w_2$, $\ldots$, $w_{c-2} \mathcal{R} w_{c-1}$ and $w_0 \models A_{1,b_1} \land \ldots \land A_{m,b_m}$.
With (2),(3),(4) together with $G(b_1, \ldots, b_m)$ follows that $\langle w_0, \ldots, w_{c-1} \rangle$ is a path in $\mathcal{M}$ and that there are no short-cuts, i.e. $|w_0, w_{c-1}| = c$, thus $\operatorname{diam}(\mathcal{M}) \geq c$.

7.2.7 LEMMA $\prod_{i=1}^{m}(b_i)$

$\forall e \in \{1, 2, \ldots\} : \exists m, b_1, \ldots, b_m \in \{2, 3, \ldots\}$ with $b_1 < \ldots < b_m$ such that:

$$G(b_1, \ldots, b_m) \quad \text{and} \quad \prod_{i=1}^{m}(b_i) > {b_m}^{4e}$$

Proof

Let prime_i be the ith prime number ($\text{prime}_1 = 2$) and choose $m := 8e$. We set $a := \prod_{i=1}^{m}(\text{prime}_i)$ and for all $i \in \{1, \ldots, m\}$: $b_i := a + \text{prime}_i$.

Assume that c prime, $c|b_i$, $c|b_j$, where $i, j \in \{1, \ldots, m\}$ and $i < j$.

Then: $c|((a + \text{prime}_j) - (a + \text{prime}_i)) \Rightarrow c|(\text{prime}_j - \text{prime}_i) \Rightarrow c < \text{prime}_j \Rightarrow c|a$. With $c|b_j$ follows $c|\text{prime}_j$, and since $c < \text{prime}_j$ we have $c = 1$. Contradiction. Thus $G(b_1, \ldots, b_m)$ is satisfied.

Finally $\prod_{i=1}^{m}(b_i) > a^m = (a^2)^{\frac{m}{2}} > (2a)^{\frac{m}{2}} = (2a)^{4e} > (a + \text{prime}_m)^{4e} = {b_m}^{4e}$.

7.2.8 LEMMA length of $T_{\langle b_1,\ldots,b_m\rangle}$

If $m, b_1, \ldots, b_m \in \{2, 3, \ldots\}$, $b_1 < \ldots < b_m$, $G(b_1, \ldots, b_m)$, then $\text{length}(P) + \text{length}(T_{\langle b_1,\ldots,b_m\rangle}) < {b_m}^4$.

Proof

If $b_m \geq 5$, then:

$$\begin{aligned}
&\text{length}(P) + \text{length}(T_{\langle b_1,\ldots,b_m\rangle})\\
&\quad= 1 + 3 + \textstyle\sum_{i=1}^{m}(2 + \text{length}(A_{i,b_i})) + (1 + \sum_{i=1}^{m}(2 + \text{length}(A_{i,b_i})))\\
&\quad< 6 + 3m + 2m\,\text{length}(A_{m,b_m})\\
&\quad= 6 + 3m + 2m(\textstyle\sum_{i=1}^{b_m}(i + 3) - 2)\\
&\quad= 6 + 3m + m({b_m}^2 + 7b_m + 10)\\
&\quad< 6 + 3b_m + b_m({b_m}^2 + 7b_m + 10)\\
&\quad< {b_m}^4.
\end{aligned}$$

Otherwise $m = 2$, $b_1 = 2$, $b_2 = 3$, and $\text{length}(P) + \text{length}(T_{\langle b_1,\ldots,b_m\rangle}) = 51 < {b_m}^4$.

7.2.9 THEOREM non-existence of standard embedding

There exists no function $g : \text{Fml}_\mathsf{K} \times \text{Th}_\mathsf{K} \to \mathbb{N}$ such that:

- $g(A, T)$ is polynomial in $\text{length}(A) + \text{length}(T)$.
- If A is a formula, T a theory, and $C :\equiv \bigwedge_{B \in T}(B)$, then $f(A, T) :\equiv C \wedge \Box C \wedge \ldots \wedge \Box^{g(A,T)} C \to A$ is an embedding of K plus theories in K.

Proof

By combining the four lemmas we can conclude that $\forall e \in \mathbb{N} : \exists T, A$ with $T \not\models A$:

$$(\mathsf{K}, \mathcal{M} \models T \text{ and } \mathsf{K}, \mathcal{M} \not\models A) \quad \Rightarrow \quad \text{diam}(\mathcal{M}) \geq (\text{length}(A) + \text{length}(T))^e$$

Now assume that we do backward proof search for such A and T in $(\mathsf{K} + T)^{\mathcal{S},2}$. The search will fail, and we can extract a countermodel. Thus there are branches in the search tree where the $(\Box)$ rule has been applied at least $(\text{length}(A) + \text{length}(T))^e$ times.

7.3 KT, KT + T

7.3.1 REMARK motivation for k

The accessibility relation of the KT models must be reflexive, thus $\Diamond A$ is also satisfied if A is satisfied in the actual world. The function used in [Fit88] simulates this interpretation. We extend it to theories.

7.3.2 DEFINITION k

We define inductively $\mathrm{k2} : \mathrm{Fml}_{\mathsf{KT}} \times \{0,1\} \to \mathrm{Fml}_{\mathsf{K}}$:

- $\mathrm{k2}(P,n) :\equiv P$
- $\mathrm{k2}(\neg P,n) :\equiv \neg P$
- $\mathrm{k2}(\Box A,0) :\equiv \Box\mathrm{k2}(A,0)$
- $\mathrm{k2}(\Box A,1) :\equiv \mathrm{k2}(A,1) \wedge \Box\mathrm{k2}(A,1)$
- $\mathrm{k2}(\Diamond A,0) :\equiv \mathrm{k2}(A,0) \vee \Diamond\mathrm{k2}(A,0)$
- $\mathrm{k2}(\Diamond A,1) :\equiv \Diamond\mathrm{k2}(A,1)$
- $\mathrm{k2}(A \wedge B,n) :\equiv \mathrm{k2}(A,n) \wedge \mathrm{k2}(B,n)$
- $\mathrm{k2}(A \vee B,n) :\equiv \mathrm{k2}(A,n) \vee \mathrm{k2}(B,n)$
- $\mathrm{k2}(A \to B,n) :\equiv \mathrm{k2}(\neg A \vee B,n)$
- $\mathrm{k2}(A \leftrightarrow B,n) :\equiv \mathrm{k2}((A \to B) \wedge (B \to A),n)$

Now we define $\mathrm{k} : \mathrm{Fml}_{\mathsf{K}} \to \mathrm{Fml}_{\mathsf{K}}$ as follows:

$$\mathrm{k}(A) :\equiv \mathrm{k2}(A,0)$$

7.3.3 DEFINITION $(\mathrm{k}+\mathrm{T})$

$$(\mathrm{k}+\mathrm{T})(A,[C_1,\dots,C_n]) := \langle \mathrm{k}(A),[\neg\mathrm{k}(\neg C_1),\dots,\neg\mathrm{k}(\neg C_n)]\rangle$$

7.3.4 THEOREM k and $(\mathrm{k}+\mathrm{T})$ are mappings

We set $(\mathrm{k}+\mathrm{T})(A,T) := \langle A',T'\rangle$. Then k is a mapping of KT in K, and $(\mathrm{k}+\mathrm{T})$ is a mapping of KT plus theories in K plus theories. We thus have:

$$\begin{aligned} \mathsf{KT} \models A \quad &\Leftrightarrow \quad \mathsf{K} \models \mathrm{k}(A) \\ \mathsf{KT}+T \models A \quad &\Leftrightarrow \quad \mathsf{K}+T' \models A' \end{aligned}$$

Proof

We prove with inductions on proof length that $\mathsf{K}^{\mathcal{S}} \vdash \mathrm{k}(A) \Leftrightarrow \mathsf{KT}^{\mathcal{S}} \vdash \epsilon \mid A$ and $(\mathsf{K}+T')^{\mathcal{S}} \vdash \epsilon \mid \mathrm{k}(A) \Leftrightarrow (\mathsf{KT}+T)^{\mathcal{S}} \vdash \epsilon \mid A$ and then use the appropriate equivalences from the chapters 4 and 5.

Example

Let A be the formula $\Diamond\neg p_0 \vee p_0$. Then $\mathsf{KT}^{\mathcal{S},2} \vdash \epsilon \mid A$, but $\mathsf{K}^{\mathcal{S},2} \nvdash \epsilon \mid A$.

We have $\mathrm{k}(A) \equiv (\Diamond\neg p_0 \vee \neg p_0) \vee p_0$, and obviously $\mathsf{K}^{\mathcal{S},2} \vdash \epsilon \mid (\Diamond\neg p_0 \vee \neg p_0) \vee p_0$.

7.3.5 THEOREM k and $(\mathrm{k}+\mathrm{T})$ are not embeddings

k and $(\mathrm{k}+\mathrm{T})$ are not embeddings.

Proof

Of course the equivalences can cause problems. But even if we consider only formulas without $\leftrightarrow$ the functions k and $(\mathrm{k}+\mathrm{T})$ are not embeddings. To show this we define $A_n :\equiv \Diamond^n p_0$.

Then $\text{length}(A_0) = 1$ and $\text{length}(\text{k}(A_{n+1})) = \text{length}(\Diamond\text{k}(A_n) \vee \text{k}(A_n)) > 2\,\text{length}(\text{k}(A))$. Thus $\text{length}(\text{k}(A_{n+1})) > 2^n$.

7.3.6 REMARK k, (k + T), and structure sharing

If no $\leftrightarrow$ occur in A and T, then the computation of $\text{k}(A)$ and $(\text{k} + \text{T})(A, T)$ is possible in polynomial time if we use structure sharing.

7.4 S4

7.4.1 REMARK S4 in KT plus theories

First we give the standard mapping of S4 in K plus theories, where we simply take certain instances of the formulas T and 4 defined in definition 1.a. (We will use this method again in chapter 11). If we omit the instances of T, then we obtain a mapping of S4 in KT plus theories.

Afterwards we construct an embedding of S4 in KT plus theories. The embedding is in fact similar to the mapping, but relies on a different motivation. If we compare backward proof search in $\mathsf{KT}^{\mathcal{S}}$ and $\mathsf{S4}^{\mathcal{S}}$, then we see that the $(\Box)$ rules handle the $\Diamond$ formulas differently. We introduce new variables for the $\Diamond$ formulas, and describe in the theory what should be done with these variables. This embedding shows that KT plus theories is quite powerful.

7.4.2 DEFINITION $(\text{k} + \text{T})_{\text{inst}}$

Let A be a formula, and let $C_1, \ldots, C_n$ be the $\Diamond$ formulas in subfmls(nnf(A)). We set

$$\begin{array}{rcl} D_1 & :\equiv & C_1 \\ D_2 & :\equiv & \phantom{C_1 \vee{}} C_2 \\ D_3 & :\equiv & C_1 \vee C_2 \\ D_4 & :\equiv & \phantom{C_1 \vee C_2 \vee{}} C_3 \\ D_5 & :\equiv & C_1 \phantom{{}\vee C_2} \vee C_3 \\ & \vdots & \\ D_{2^n-1} & :\equiv & C_1 \vee \ldots \vee C_n \end{array}$$

i.e. the formulas $D_1, \ldots, D_{2^n-1}$ are all possible disjunctions formed of the formulas $C_1, \ldots, C_n$. Finally $T' := \text{T}\{C_1/p_0\}, \ldots, \text{T}\{C_n/p_0\}, 4\{D_1/p_0\}, \ldots, 4\{D_{2^n-1}/p_0\}$.

Then:

$$(\text{k} + \text{T})_{\text{inst}}(A) := \langle A, T' \rangle$$

Example

Let A be the formula $\Diamond(\Diamond p_1 \vee p_2) \wedge p_0$. Then the $\Diamond$ formulas in subfmls(nnf(A)) are $\Diamond(\Diamond p_1 \vee p_2)$ and $\Diamond p_1$. Thus $C_1 \equiv \Diamond(\Diamond p_1 \vee p_2)$, $C_2 \equiv \Diamond p_1$, $D_1 \equiv \Diamond(\Diamond p_1 \vee p_2)$, $D_2 \equiv \Diamond p_1$, and $D_3 \equiv \Diamond(\Diamond p_1 \vee p_2) \vee \Diamond p_1$. We obtain:

$$\begin{array}{rcl} T' & = & \Box\Diamond(\Diamond p_1 \vee p_2) \to \Diamond(\Diamond p_1 \vee p_2), \\ & & \Box\Diamond p_1 \to \Diamond p_1, \\ & & \Box\Diamond(\Diamond p_1 \vee p_2) \to \Box\Box\Diamond(\Diamond p_1 \vee p_2), \\ & & \Box\Diamond p_1 \to \Box\Box\Diamond p_1, \\ & & \Box(\Diamond(\Diamond p_1 \vee p_2) \vee \Diamond p_1) \to \Box\Box(\Diamond(\Diamond p_1 \vee p_2) \vee \Diamond p_1) \end{array}$$

For a second example see remark 11.3.2.

7.4.3 THEOREM $(\mathrm{k}+\mathrm{T})_{\mathrm{inst}}$ is a mapping

If $(\mathrm{k}+\mathrm{T})_{\mathrm{inst}}(A) = \langle A', T'\rangle$, then:

$$\mathsf{S4} \models A \quad \Leftrightarrow \quad \mathsf{K} + T' \models A'$$

Proof

'$\Rightarrow$':

We take a proof $\mathcal{P}$ of nnf(A) in $\mathsf{S4}^{\mathcal{S}}$ and show that we can convert it into a proof $\mathcal{P}'$ of nnf(A) in $(\mathsf{K}+\mathrm{nnf}(T'))^{\mathcal{S}}$. The theorem then follows with the appropriate equivalences from the chapters 4 and 5.

We start at the root of $\mathcal{P}$ and move towards the leaves. The root of $\mathcal{P}'$ is the sequent nnf(A), nnf$(\neg T')$. Applications of $(\wedge)$ and $(\vee)$ in $\mathcal{P}$ are converted into the corresponding applications of $(\wedge)$ and $(\vee)$ in $\mathcal{P}'$. The following two proof fragments show how applications of $(\Diamond)$ are simulated in $(\mathsf{K}+\mathrm{nnf}(T'))^{\mathcal{S}}$. Note that $\mathrm{nnf}(\neg\mathrm{T}\{\neg A/p_0\}) \equiv \mathrm{nnf}(\neg(\Box\neg A \to \neg A)) \equiv \Box\mathrm{nnf}(\neg A) \wedge \mathrm{nnf}(A)$.

$$\dfrac{\Diamond\mathrm{nnf}(A), \Diamond\Sigma \mid \mathrm{nnf}(A), \Gamma}{\Diamond\Sigma \mid \Diamond\mathrm{nnf}(A), \Gamma}(\Diamond) \quad\rightsquigarrow\quad \dfrac{\dfrac{\Diamond\mathrm{nnf}(A), \Diamond\Sigma \mid \Box\mathrm{nnf}(\neg A), \Gamma \quad \Diamond\mathrm{nnf}(A), \Diamond\Sigma \mid \mathrm{nnf}(A), \Gamma}{\Diamond\mathrm{nnf}(A), \Diamond\Sigma \mid \Box\mathrm{nnf}(\neg A) \wedge \mathrm{nnf}(A), \Gamma}(\wedge)}{\Diamond\Sigma \mid \Diamond\mathrm{nnf}(A), \Box\mathrm{nnf}(\neg A) \wedge \mathrm{nnf}(A), \Gamma}(\Diamond)$$

In a similar way we simulate $(\Box)$ applications. We assume that D is the disjunction from definition 7.4.2 that corresponds to the multiset $\Diamond\Sigma$. Note that $\mathrm{nnf}(\neg 4\{\neg D/p_0\}) \equiv \mathrm{nnf}(\neg(\Box\neg D \to \Box\Box\neg D)) \equiv \Box\mathrm{nnf}(\neg D) \wedge \Diamond\Diamond\mathrm{nnf}(D)$.

$$\dfrac{\Diamond\Sigma \mid \mathrm{nnf}(A), \Diamond\Sigma}{\Diamond\Sigma \mid \Box\mathrm{nnf}(A), \Gamma}(\Box) \quad\rightsquigarrow\quad \dfrac{\dfrac{\epsilon \mid \mathrm{nnf}(\neg D), \Sigma}{\Diamond\Sigma \mid \Box\mathrm{nnf}(A), \Box\mathrm{nnf}(\neg D), \Gamma}(\Box) \quad \dfrac{\dfrac{\epsilon \mid \mathrm{nnf}(A), \Diamond\mathrm{nnf}(D), \Sigma}{\Diamond\Diamond\mathrm{nnf}(D), \Diamond\Sigma \mid \Box\mathrm{nnf}(A), \Gamma}(\Box)}{\Diamond\Sigma \mid \Box\mathrm{nnf}(A), \Diamond\Diamond\mathrm{nnf}(D), \Gamma}(\Diamond)}{\Diamond\Sigma \mid \Box\mathrm{nnf}(A), \Box\mathrm{nnf}(\neg D) \wedge \Diamond\Diamond\mathrm{nnf}(D), \Gamma}(\wedge)$$

'$\Leftarrow$':

Obviously the formulas provable in $(\mathsf{K}+T')^{\mathcal{H}}$ are a subset of the formulas provable in $\mathsf{S4}^{\mathcal{H}}$. The theorem follows with the appropriate equivalences from the chapters 4 and 5.

7.4.4 THEOREM $(\mathrm{k}+\mathrm{T})_{\mathrm{inst}}$ is not an embedding

The result of $(\mathrm{k}+\mathrm{T})_{\mathrm{inst}}(A)$ can have exponential length (relative to length(A)).

Proof

If there are n $\Diamond$ formulas in subfmls(nnf(A)), then the theory in $(\mathrm{k}+\mathrm{T})_{\mathrm{inst}}(A)$ contains $2^n - 1$ instances of the formula 4.

7.4.5 REMARK change $(\mathrm{k}+\mathrm{T})_{\mathrm{inst}}$ to an mapping in KT plus theories

We could easily make an mapping of $\mathsf{S4}$ in KT plus theories out of the mapping $(\mathrm{k}+\mathrm{T})_{\mathrm{inst}}$ by omitting the instances of the formula T. However, this would still not be an embedding, since the instances of the formula 4 and not those of T are the reason of the problem.

7.4.6 DEFINITION extended substitution

We extend the notation for substitution:

Let A be a formula, $Q_1, \ldots, Q_n$ pairwise different variables that do not occur in A, and $C_1, \ldots, C_n$ subformulas of A.

We construct a function $\pi : \{1, \ldots, n\} \to \{1, \ldots, n\}$ such that $\pi(i) < \pi(j)$ implies $\text{length}(C_{\pi(i)}) \geq \text{length}(C_{\pi(j)})$. First we substitute in A the variable $Q_{\pi(1)}$ for $C_{\pi(1)}$, in the result we substitute the variable $Q_{\pi(2)}$ for $C_{\pi(2)}$, $\ldots$, and finally the variable $Q_{\pi(n)}$ for $C_{\pi(n)}$. We write $A\{Q_1/C_1, \ldots, Q_n/C_n\}$ for the result.

Example

If $A \equiv \Diamond(p_0 \vee \Diamond\Box p_1) \wedge \Diamond\Box p_1$ and $C_1 \equiv \Diamond\Box p_1$, $C_2 \equiv \Diamond(p_0 \vee \Diamond\Box p_1)$, then $\pi(1) = 2$, $\pi(2) = 1$, and $A\{Q_1/C_1, Q_2/C_2\} \equiv Q_2 \wedge Q_1$.

7.4.7 DEFINITION (kt + T)

Assume that:

- A is a formula.
- $\Diamond C_1, \ldots, \Diamond C_n$ are the $\Diamond$ formulas in $\text{subfmls}(A)$.
- $Q_1, \ldots, Q_n$ are variables that do not occur in A.
- For all $i \in \{1, \ldots n\} : C_i' :\equiv C_i\{Q_1/\Diamond C_1, \ldots, Q_n/\Diamond C_n\}$.
- $A' :\equiv A\{Q_1/\Diamond C_1, \ldots, Q_n/\Diamond C_n\}$
- $T' := \{Q_1 \vee \neg(\Diamond Q_1 \vee \Diamond C_1'), \ldots, Q_n \vee \neg(\Diamond Q_n \vee \Diamond C_n')\}$.

Then:

$$(\text{kt} + \text{T})(A) := \langle A', T' \rangle$$

Example

Assume that $A \equiv \Diamond(p_0 \vee \Box\Diamond\neg p_0) \wedge p_0$.

Thus $n = 2$, $C_1 \equiv p_0 \vee \Box\Diamond\neg p_0$, $C_2 \equiv \neg p_0$. We choose $Q_1 \equiv p_1$, $Q_2 \equiv p_2$. The variables Q_1, Q_2 are used as abbreviations for the $\Diamond$ formulas C_1, C_2.

Then $C_1' \equiv p_0 \vee \Box Q_2$, $C_2' \equiv \neg p_0$, $A' \equiv Q_1 \wedge p_0$, and $T' = \{Q_1 \vee \neg(\Diamond Q_1 \vee \Diamond C_1'), Q_2 \vee \neg(\Diamond Q_2 \vee \Diamond C_2')\}$.

7.4.8 THEOREM (kt + T) is an embedding

If $(\text{kt} + \text{T})(A) = \langle A', T' \rangle$, then $\text{length}(A') + \text{length}(T')$ is polynomial (relative to $\text{length}(A) + \text{length}(T)$), and

$$\mathsf{S4} \models A \quad \Leftrightarrow \quad \mathsf{KT} + T' \models A'$$

Proof

The first statement is obvious. In order to prove the second statement, we prove

$$\mathsf{S4}^{\mathcal{S}} \vdash \epsilon \mid \text{nnf}(A) \quad \Leftrightarrow \quad (\mathsf{KT} + \text{nnf}(T'))^{\mathcal{S}} \vdash \epsilon \mid \text{nnf}(A')$$

With the appropriate equivalences from the chapter 4 and 5 we obtain the theorem.

'$\Rightarrow$':

Let $\mathcal{P}$ be a proof of $\epsilon \mid \text{nnf}(A)$ in $\mathsf{S4}^{\mathcal{S}}$. We construct a proof $\mathcal{P}'$ of $\text{nnf}(A')$ in $(\mathsf{KT} + \text{nnf}(T'))^{\mathcal{S}}$, starting at the root and moving towards the leaves. The root of $\mathcal{P}'$ is $\text{nnf}(A'), \text{nnf}(\neg T')$. An application of one of the rules ($\vee$), ($\wedge$), ($\Box$) in $\mathcal{P}$ becomes an application of the corresponding rule of $(\mathsf{KT}+\mathrm{nnf}(T'))^{\mathcal{S}}$ in $\mathcal{P}'$. The only problem is caused by the ($\Diamond$) rule. If this rule is applied backwards with main formula $\Diamond C_i$ in $\mathcal{P}$, then we only have a Q_i in the place of the $\Diamond C_i$ in $\mathcal{P}'$. Therefore we insert the following part in $\mathcal{P}'$ as soon as a Q_i occurs in a sequent.

$$\dfrac{\dfrac{}{\Diamond\Sigma \mid Q_i, \neg Q_i, \Gamma}\,(\mathrm{id}) \qquad \dfrac{\dfrac{\Diamond Q_i, \Diamond\Sigma \mid Q_i, \Diamond\mathrm{nnf}(C_i'), \Gamma}{\Diamond\Sigma \mid Q_i, \Diamond Q_i, \Diamond\mathrm{nnf}(C_i'), \Gamma}\,(\Diamond)}{\Diamond\Sigma \mid Q_i, \Diamond Q_i \vee \Diamond\mathrm{nnf}(C_i'), \Gamma}\,(\vee)}{\Diamond\Sigma \mid Q_i, \neg Q_i \wedge (\Diamond Q_i \vee \Diamond\mathrm{nnf}(C_i')), \Gamma}\,(\wedge)$$

The inserted part has two purposes:

- If $\Diamond\mathrm{nnf}(C_i)$ occurs for the first time on a branch in $\mathcal{P}$ (seen from the root), then it makes sure that $\Diamond\mathrm{nnf}(C_i')$ occurs in the corresponding sequent in $\mathcal{P}'$.
- In contrast to $\mathsf{S4}^{\mathcal{S}}$, a backward application on ($\Box$) in $(\mathsf{KT}+T)^{\mathcal{S}}$ destroys leading $\Diamond$ connectives. The inserted part regenerates them by 'replacing' Q_i by $\Diamond Q_i$.

Thus we know that if $\Diamond\mathrm{nnf}(C_i)$ occurs in a sequent in $\mathcal{P}$, then $\Diamond\mathrm{nnf}(C_i')$ occurs in the corresponding sequent in $\mathcal{P}'$, i.e. we can apply ($\Diamond$) backwards in $\mathcal{P}'$ and continue with the construction.

'$\Leftarrow$':

Let $\mathcal{P}'$ be a proof of $\mathrm{nnf}(A')$ in $(\mathsf{KT}+\mathrm{nnf}(T'))^{\mathcal{S}}$. We construct a proof $\mathcal{P}$ of $\mathrm{nnf}(A)$ in $\mathsf{S4}^{\mathcal{S}}$, starting at the root and moving towards the leaves. The root of $\mathcal{P}$ is $\mathrm{nnf}(A)$.

First some remarks about the proof $\mathcal{P}'$. We can assume that the ($\wedge$) rule is only applied backwards on a formula from T' if the corresponding Q_i occurs in the sequent (for the first time since the last backward application of ($\Box$) on this branch). Because of the invertibility of ($\vee$), ($\wedge$), and ($\Diamond$) we can moreover assume that if ($\wedge$) is applied backwards on a formula from T', then ($\vee$) and ($\Diamond$) are applied backwards as in the proof fragment above.

Now the translation of $\mathcal{P}'$ into a proof $\mathcal{P}$ in $\mathsf{S4}^{\mathcal{S}}$ causes no problems. We simply do nothing in $\mathcal{P}$ if the proof fragment from above occurs in $\mathcal{P}'$.

7.5 $\mathsf{S4}+T$

7.5.1 REMARK motivation

Since the accessibility relation in the $\mathsf{S4}$ models is transitive, $\Box B$ means that B is true in the current world and in all the worlds in the future.

7.5.2 THEOREM S4 plus theories in S4

$$\mathsf{S4}+B_1,\ldots,B_n \models A \quad\Leftrightarrow\quad \mathsf{S4}\models \Box B_1 \wedge \ldots \wedge \Box B_n \to A$$

Proof

See the proof of theorem 3.7.3.

7.6 OTHER LOGICS

7.6.1 THEOREM K plus theories in CTL

We define inductively ctl_1 for formulas in negation normal form: $\mathrm{ctl}_1(P) :\equiv P$, $\mathrm{ctl}_1(\neg P) :\equiv \neg P$, $\circ \in \{\wedge, \vee\} \Rightarrow \mathrm{ctl}_1(B \circ C) :\equiv \mathrm{ctl}_1(B) \circ \mathrm{ctl}_1(C)$, $\mathrm{ctl}_1(\Box B) :\equiv \mathsf{AX}\,\mathrm{ctl}_1(B)$, $\mathrm{ctl}_1(\Diamond B) :\equiv \mathsf{EX}\,\mathrm{ctl}_1(B)$. If A is in negation normal form, then:

$$\mathsf{K} + B_1, \ldots, B_n \models A \quad \Leftrightarrow \quad \mathsf{CTL} \models \mathsf{AG}\,B_1 \wedge \ldots \wedge \mathsf{AG}\,B_n \to \mathrm{ctl}_1(A)$$

7.6.2 THEOREM S4 in CTL

We define inductively ctl_2 for formulas in negation normal form: $\mathrm{ctl}_2(P) :\equiv P$, $\mathrm{ctl}_2(\neg P) :\equiv \neg P$, $\circ \in \{\wedge, \vee\} \Rightarrow \mathrm{ctl}_2(B \circ C) :\equiv \mathrm{ctl}_2(B) \circ \mathrm{ctl}_2(C)$, $\mathrm{ctl}_2(\Box B) :\equiv \mathsf{AG}\,\mathrm{ctl}_2(B)$, $\mathrm{ctl}_2(\Diamond B) :\equiv \mathsf{EF}\,\mathrm{ctl}_2(B)$. If A is in negation normal form, then:

$$\mathsf{S4} \models A \quad \Leftrightarrow \quad \mathsf{CTL} \models \mathrm{ctl}_2(A)$$

7.6.3 THEOREM S4 in S5

$$\mathsf{S4} \models A \quad \Leftrightarrow \quad \mathsf{S5} \models \Diamond\Box A$$

7.6.4 REMARK modal logics in classical predicate logic

See [Ohl91] and [OS97] for embeddings of modal logics in classical predicate logic. In [HS97] an implementation for K is discussed.

7.6.5 REMARK IPC in S4

There exist several embeddings of IPC in S4 (see for example [TS96]). See [Dos93] for embeddings of IPC in other modal logics, and also for embeddings of substructural logics in modal logics.

7.6.6 REMARK S4 in linear logic

There exists an embedding of S4 in linear logic (for formulas without $\leftrightarrow$). See for example [MM94] for more information.

7.7 SUMMARY

We have first defined the notions of embedding (where the length of the output is polynomial, relative to the length of the input) and of mapping (no such restriction). Then we have given the obvious mappings of KT in K, $\mathsf{KT} + T$ in $\mathsf{K} + T$, and $\mathsf{K} + T$ in K. Moreover we have proved that there is no 'standard' embedding of $\mathsf{K} + T$ in K. For S4 we have discussed both a mapping in $\mathsf{KT} + T$ (using instances of the formulas T, 4) and an embedding in $\mathsf{KT} + T$ (using abbreviations).

II

IMPLEMENTATION

8

THE LWB PROJECT

1994 Range Rover LWB County
— Available for: $36,900 —

4 Door, S.U.V., Automatic Transmission, Power ABS Brakes, Power Asst. Steering, Heated/Leather/Power Seats, Power Windows, Power Door Locks, Rear Defrost, Rear Wiper, Tilt Steering, Cruise Control, Inter. Wipers, Tinted Glass, Roof Rack, Security Alarm, Cassette, Power Mirrors. 35,862 Miles, V.I.N.# 645329.

8.1 INTRODUCTION

The Logics Workbench (LWB) is being developed at the university of Bern. The aim is to facilitate the access to and the use of a variety of propositional logical formalisms. Considerable effort has been made to produce a user-friendly software (graphical user interface, information system, ...).

The LWB includes implementations of the decision procedures discussed in part I of this thesis. Version 1.0, which is described in the following, was used for all computations in part III.

8.2 OVERVIEW

The LWB consists of five components:

- Graphical user interface.
- Parser/interpreter.
- Logic modules.
- Kernel.
- Information system.

The graphical user interface has been developed in order to facilitate the handling of sessions, the navigation within the program, and the reuse of previously typed statements. However, it is also possible to use the LWB in a shell via the ASCII interface.

The parser/interpreter obtains its input from the user interface, translates and executes it. The basic communication with the system is done in an untyped language, which follows very closely the philosophy of well-established computer algebra systems such as Maple and Mathematica. It includes a full programming language.

The logic modules contain a collection of implementations of well-known algorithms of the most important propositional logics:

- classical propositional logic (CPC)
- intuitionistic propositional logic (IPC)
- modal propositional logics (K, KT, $\mathsf{S4}$, $\mathsf{S5}$, G)
- multimodal propositional logics (K_n, KT_n, $\mathsf{S4}_n$)
- tense and temporal propositional logics (K_t, PLTL)
- linear propositional logic
- nonmonotonic propositional logics (autoepistemic reasoning, circumscription, closed world assumption, default logic)

The functions that are available for a typical monotonic logic module are grouped as follows:

- Conversions: Remove true and false, remove provable subformulas, replace $\rightarrow$ by $\neg\vee$, simplify a formula (only for CPC), ...
- General functions: Length of a formula, list of the variables that occur in a formula, ...
- Miscellaneous: Embeddings, random formulas, standard formulas.
- Normal forms: Compute normal forms of formulas, check whether a formula is in a certain normal form.
- Proofs: Compute a sequent calculus proof (only for CPC).
- Provability: Check whether a formula is provable, whether a formula is derivable from a theory, whether a theory is consistent.

Note that the user can add his own functions (implemented in the LWB programming language), but that he cannot add 'logics'. There is for example no generic procedure that does proof search in a sequent calculus given by the user.

The LWB kernel provides the central data structures on which both the parser and the logic modules rely. This includes the data structures that can be created by the user (for example formulas, lists), and also data structures that are used internally.

A comprehensive information system gives detailed support for the usage of the different modules. All information is available via the World Wide Web and may therefore be consulted without any local installations.

8.3 COMPARISON WITH OTHER SYSTEMS

The system with the most similarities with the LWB is probably the Helsinki Logic Machine (HLM) (see [NT87]). The main difference between the LWB and the HLM is probably that more man-years have been invested in the LWB project. This allowed us to develop a user-friendly interface. We cannot compete with the large number of decision procedures in the HLM, since each decision procedure in the LWB has been implemented separately, while in the HLM in many cases embeddings in dynamic logic were used. On the other hand, our solution gives us the possibility to implement efficient decision procedures by making use of properties of the specific logics.

The following list should help to make it clear what the LWB is not:

- The LWB is not a theorem prover.

  First, we offer a large variety of functions, not just decision procedures. Second, we do not offer possibilities to prove a formula in several steps (for example using lemmas), and there is no way the user could guide the search.

  We do not try to list existing theorem provers, since there are simply too many.

- The LWB is not a tool to learn the construction of proofs.

  The proofs in the LWB are built up automatically. The user can display the proofs in several ways, and he can go step by step through the proofs, but he cannot change them or construct them himself.

  See Jape [BS97], Mac Logic [Dyc91], or Proof Tutor [SS92] if you want to learn how to construct proofs.

- The LWB does not rely on a logical framework.

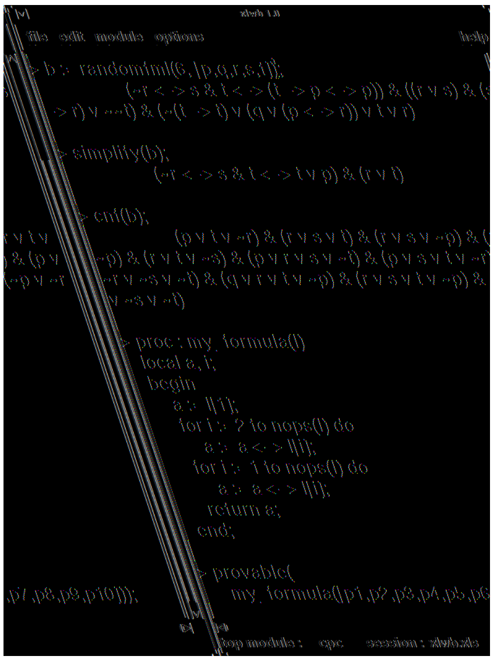
  There is a considerable number of logic modules in the LWB, and each one contains many functions. The user can add his own functions with the LWB programming language. However, there is no way to define a logic, for example by giving a sequent calculus, and to use a generic decision procedure for proof search in this logic. This is certainly a restriction. We chose this solution because it seemed to be too difficult for us to define a framework that comprises all the logics that are contained in the LWB (think for example of linear logic, PLTL, or the calculus $\mathsf{S4}^{\mathcal{S},2}$) without becoming very inefficient.

  See for example Isabelle [Pau94] if you are looking for a logical framework.

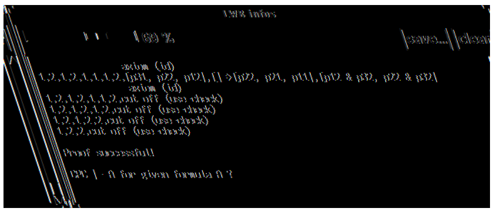
8.4 KERNEL

Data structures for formulas and other expressions that are used in algorithms are represented in the kernel of the LWB. The principles of object oriented design have been applied. Care was taken to provide the developers of LWB modules with a clear, abstracted interface to the basic objects. C++ allows good encapsulation, for example pointer handling becomes superfluous for the implementation of algorithms. Moreover, we took advantage of the possibility of operator overloading which makes dealing with expressions more intuitive and increases the readability of functions.

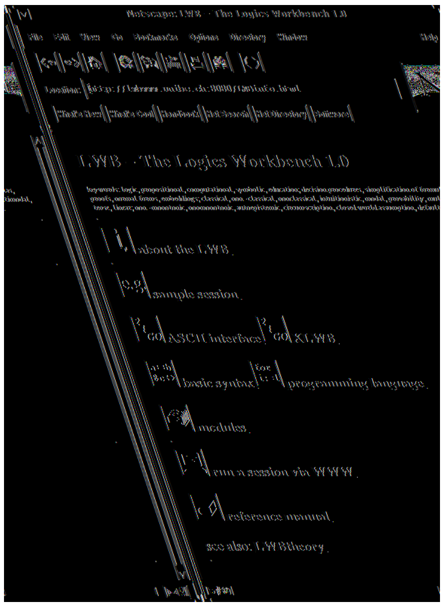
Great importance has been attached to a clear design of the internal class hierarchy and high efficiency of the basic member functions. The addition of new data types is simple; thanks to the inheritance mechanism of classes, only those parts that really differ from the other data types need to be implemented. Moreover, there is a close relationship to the parser: the nonterminals of the grammar of the parser correspond to the abstract base classes, and the terminals correspond to the derived concrete classes. Reference counting has been chosen as internal memory management.


xlwb 1.0
file edit module options help
top module : cpc session : xlwb.xls

```
> b := randomfml(6,[p,q,r,s,t]);
              (~r <-> s & t <-> (t -> p <-> p)) & ((r v s) & (
   -> r) v ~~t) & (~(t -> t) v (q v (p <-> r)) v t v r)

> simplify(b);
              (~r <-> s & t <-> t v p) & (r v t)

> cnf(b);
              (p v t v ~r) & (r v s v t) & (r v s v ~p) & (
   -p) & (r v t v ~s) & (p v r v s v ~t) & (p v s v t v ~r
   ~r v ~s v ~t) & (q v r v t v ~p) & (r v s v t v ~p) &
   v ~s v ~t)

> proc : my_formula(l)
  local a, i;
  begin
     a := l[1];
     for i := 2 to nops(l) do
        a := a <-> l[i];
     for i := 1 to nops(l) do
        a := a <-> l[i];
     return a;
  end;

> provable(
     my_formula([p1,p2,p3,p4,p5,p6,p7,p8,p9,p10]));
```



```
              axiom (id)
1,2,1,2,1,1,1,2,[p31, p22, p12],[]->[p22, p21, p11],[p12 & p32, p22 & p32]
              axiom (id)
1,2,1,2,1,1,2,cut off (use check)
1,2,1,2,1,2,cut off (use check)
1,2,1,2,2,cut off (use check)
1,2,2,cut off (use check)

Proof successfull!

CPC |- A for given formula A ?
```


Netscape: LWB - The Logics Workbench 1.0
File Edit View Go Bookmarks Options Directory Window Help
Location: http://lwbwww.unibe.ch:8080/LWBinfo.html
What's New What's Cool Handbook Net Search Net Directory Software

LWB - The Logics Workbench 1.0

keywords: logic, propositional, computational, symbolic, education, decision procedures, simplification of formulas, proofs, normal forms, embeddings, classical, non-classical, nonclassical, intuitionistic, modal, provability, ... tense, linear, non-monotonic, nonmonotonic, autoepistemic, circumscription, closed world assumption, default ...

about the LWB.

sample session.

GO ASCII interface GO XLWB.

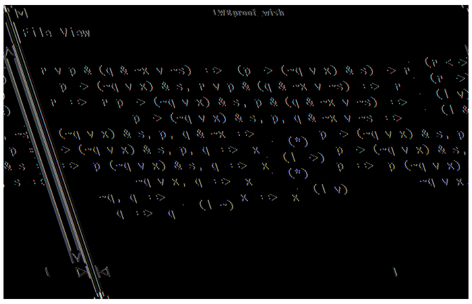
basic syntax programming language.

modules.

run a session via WWW.

reference manual.

see also: LWBtheory.



```
r v p & (q & ~x v ~s) :=> (p -> (~q v x) & s) -> r    (r <->
  p -> (~q v x) & s, r v p & (q & ~x v ~s) :=> r      (r ->
r :=> r p -> (~q v x) & s, p & (q & ~x v ~s) :=>      (l v)
           p -> (~q v x) & s, p, q & ~x v ~s :=>        (l &
   (~q v x) & s, p, q & ~x :=>       (*)  p -> (~q v x) & s, p
   -> (~q v x) & s, p, q :=> x     (l ->)  p -> (~q v x) & s,
   :=> p (~q v x) & s, q :=> x      (*)   p :=> p (~q v x)
           ~q v x, q :=> x            (l v)              ~q v x
       ~q, q :=>     (l ~) x :=> x
        q :=> q
```

Figure 8.a: The main window of the LWB (top left), the info window (bottom left), LWBinfo (top right), proof_wish (bottom right).

8.5 PARSER/INTERPRETER

To set up a user-friendly parser we used the well-known development tools *lex* and *yacc*. With *lex*, the lexical analyser is generated (recognition of tokens) and with *yacc*, the actual parser is built. The description of the accepted language is given in Backus-Naur Form. Both tools translate the intuitive descriptions into C programs. The parser is designed to respect the usual conventions in the writing of expressions.

The LWB programming language gives the user the possibility to define new procedures. The programming language offers constructs like `for to do`, `while do`, `foreach do`, data structures like lists, arrays, and procedures with parameters, reference variables, local variables, and local procedures.

8.6 LOGIC MODULES

Each logic module provides a series of functions. These functions are implemented in C++. In order to structure the description of the logic modules, we distinguish between monotonic propositional systems and nonmonotonic propositional systems.

8.6.1 Monotonic propositional logics

The largest group of implemented modules concerns monotonic propositional logics. The LWB tries to cover a great variety of propositional logics (including the most important ones), but it is not our aim to include every system which is described in the literature.

In order to facilitate queries to the LWB, we do not require that the input be in normal form. Even a seemingly simple transformation such as the step from $A \leftrightarrow B$ to $(A \to B) \land (B \to A)$ can convert a well structured formula into a completely unreadable one.

Auxiliary functions: There are auxiliary functions (such as length and depth of formulas), standard functions for the computation of normal forms, embeddings, ... There is the possibility to generate 'random formulas', depending on given parameters like depth or list of variables. Of course the notion of random formula is problematic, but there are approximations that are sufficient for our purpose (cp. chapter 9).

Decision procedures: For each logic under consideration a specifically tailored decision procedure is used. Some of these decision procedures are explained in detail in this thesis.

We do not treat several logical calculi simultaneously by reducing them to one specific base system. For example, there exists a straightforward embedding of the modal logic KT into K (see chapter 7). Although the embedding is very simple, nested modal operators and equivalences can cause an exponential increase of the length of the formula. If one uses structure sharing and if no equivalences occur in the formula, the problem can be solved. However, proof search in K for the embedded formula is still considerably harder than proof search for the original formula in KT, because KT allows a much stronger form of use-check than K (cp. remark 5.2.18).

Besides pure provability, the LWB is also able to cope with provability of a formula in a logic plus theory, i.e. in a logic plus additional nonlogical axioms. Of course this is trivial for classical and intuitionistic logic, but requires some effort in the case of several modal logics. Two examples are the wise-men-problem in multimodal logic and the proof of the equivalence of different modal logics (see chapter 11).

Proofs: In the module for classical propositional logic, there is a function `proof` that generates proofs in the standard sequent calculus. At first, the usual backward proof search is used to find a proof, and then this proof is simplified in order to make it more readable.

Simplification of formulas: It is not difficult to give some definition of 'simplification' with pleasant properties. Two possibilities are the conjunctive normal form or the minimised conjunctive normal form. In a few cases, normal forms of this kind may help, but in general the results are far too long. Moreover, we think that the user should be able to recognise the structure of his input formula after the simplification process – a condition that is often ignored.

Therefore, we sacrifice the requirement to obtain a 'minimal' solution in some theoretical sense. Simplification of a formula can be described vaguely in our context as the transformation of a formula to a logically equivalent form which is better readable, yet does not destroy the original structure unnecessarily.

These requirements called for the development of a rather sophisticated algorithm. On the one hand, it tries to find as many simplifying sequences of transformations as possible. On the other hand, for efficiency reasons, it cannot use an exhaustive search to find these sequences, but relies on a very directed search. In addition, the user may control (via parameters) the tradeoff between quality and runtime of the simplification process according to his needs. See the sample session in section 8.11 for a simple example.

Efficiency: We use several methods to increase the efficiency of the implemented algorithms. None of the methods described below is completely new. However, it turned out that their combination considerably improves the runtime behaviour without leading to unintelligible algorithms.

Structure sharing is used on several occasions. On the one hand it is supported by the kernel of the LWB: If the user types `a := p v q & r;` and afterwards `b := a v p;`, then the formula `a` is not copied in order to store `b`, but simply a pointer to `a` is used. This method helps to save memory and can speed up equality tests. On the other hand most decision procedures make use of structure sharing in order to avoid the copying of the whole sequent when only the head of the sequent changes.

Use-check (cp. [SFH92]) is a widely applicable method in the context of proof search for sequent calculi. The basic idea is the following: Suppose we have to check whether $A \wedge B, \Gamma$ is provable. First we check whether A, Γ is provable. If this is the case and if A was not 'used' in the proof of A, Γ, then it is superfluous to search for a proof of B, Γ since the provability of this sequent follows immediately from that of A, Γ. See chapter 5 for details.

Another method is the efficient loop-check for example for the modal logic S4. Because the usual sequent calculus for S4 contains a hidden contraction, loops can occur during proof search. Therefore we have to test for each sequent whether it already occurred on the current branch of the search tree. Such a loop-check is in general expensive, and the implementation of an equality test for sequents is laborious. In the case of S4 there exists a calculus where loop-checking is very cheap (see definition 5.6.8). The main problem, namely the presence of noninvertible rules, still remains, but this problem is related to the PSPACE property of the validity problem for S4 and therefore, difficult to avoid.

8.6.2 Nonmonotonic propositional logics

Most of the interesting properties as well as the known problems inherent in the systems above do already show up in the propositional case, and it is sometimes even easier in this framework to construct examples which illustrate these problems. Therefore, at least for didactical purposes, the restriction to the propositional case should not be too disturbing.

Following the pattern of monotonic systems, a number of functions besides decision procedures have been implemented. A good example is the function that computes the set of extensions of a default theory. This function gives a lot more insight into default logic than the mere provability test.

In section 8.11 we use some of these functions in a simple example.

8.7 GRAPHICAL USER INTERFACE

The graphical user interface (GUI) has been developed under Motif. The GUI contains 3 components (see also figure 8.3):

1. The main window: The main window consists of a menu bar and a scrollable input window, which can be changed in size. The input window is the essential part of the GUI. It is built up by pairs of input and output regions. By returning to previous input regions the user can avoid repeated typing of equal or similar commands. Each input region is a little multiline text editor, in which all needed text manipulations can be done.
2. Status window: It displays general information about the state and the activity of the LWB and specific information about the run of algorithms, which is of interest to the user. The amount of information can be controlled.
3. Option window: In the option window the user can change global variables like display mode or bracket mode by mouse clicks. Likewise, he can choose his favourite colour settings.

Special emphasis has been put on the similarity to other X11-/Motif-applications. For example, user specific adaptations are done in the usual way (resource file). Various useful features are supported:

1. Save/load of all assigned variables, or the list of input and output regions (session).
2. Insertion and deletion of regions. Hiding the output regions. Reevaluation of all input regions.
3. Emacs-like control sequences.
4. Connection to the information system.

The proofs generated for classical propositional logic can be graphically displayed using the proof wish tool (see figure 8.3).

8.8 LWB INFORMATION SYSTEM

The LWB information system goes far beyond what is usually offered by other systems of computer assisted logics and mathematics. It relies on the features provided by World Wide Web (WWW). This approach offers several advantages:

1. Representation: The HTML format includes fonts in different shapes and sizes, but also icons and pictures.
2. Hypertext links: The LWB information system uses the capabilities of hypertext to structure the available information.
3. Availability: With a WWW Browser one can use the LWB information system installed in Bern. Users from outside Bern can also install it easily at their site.
4. Flexibility: A key feature is the easy extendability. Existing parts are not affected by changes or extensions.

The LWB information system also contains a large variety of information besides the standard help texts:

1. Information about the LWB: Aims, installation instructions, ...
2. Sample session with the LWB, picture of a sample session with the XLWB.
3. Description of syntax and programming language.
4. Description of each function (with examples), list of the functions in a module, list of the modules.

Moreover, it is possible to use the LWB installation in Bern via the information system. This facility is especially useful for occasional queries, because no preceding local installation of the LWB or of the LWB information system is required. Naturally this usage of the LWB is not very user-friendly, and the number of simultaneous users is limited.

The function descriptions of the LWB information system are automatically generated from the corresponding description in the reference manual, thus ensuring consistency. Moreover, all examples in the reference manual are automatically calculated by the LWB during typesetting and the results are inserted. Therefore, the reader cannot be confused by examples that do not correspond to the behaviour of the LWB.

8.9 THE LWB IN EDUCATION

The LWB is used at the University of Bern in connection with courses on logic and computer science. It has turned out to be helpful in considerably reducing the time spent on tedious calculations and transformations that do not give much insight, so that more time was left to concentrate on essential problems.

Using the LWB, it is possible to discuss problems of realistic size which go far beyond the usual toy examples. In this way students get a better feeling for algorithmic aspects of logic in general and especially for the complexities (run time, length of formulas, . . .) relevant in practice. A typical example is the computation of a conjunctive normal form of a formula A. There are two well-known procedures: In the first one we use the distributivity law, in the second one we introduce abbreviations for subformulas. The second one looks more complicated, and the result is not equivalent to A (we just know that A is satisfiable iff the result is satisfiable). The disadvantage of the first one is that the length of the result can be exponentially long relative to the length of A. However, it is difficult to show this on a blackboard, since it becomes only fully visible if A is not too short.

A drawback of textbooks is the often rather small number of exercises. The LWB gives the possibility to look at variations of these examples. This often helps to better understand what the essential point of the example is. In circumscription for example, we can try the effects of all combinations of minimised, varied, and 'usual' variables.

The LWB programming language gives the students the possibility to implement standard functions. Advantage (compared with other programming languages): The required data structures, input and output, . . . are integrated in the LWB.

The following examples are typical exercises:

- Complexity:
  Find a formula A such that the conjunctive normal form of A (computed using the distributivity law) is 1000 times as long as A. Check your solution with the LWB.
- Algorithms:
  Implement of a function that computes the negation normal form in classical propositional

logic. Especially for computer science students it is probably easier to understand such programs than pure mathematical definitions. A more demanding programming exercise is the drawing of diagrams (cp. chapter 14). There, the main difficulty is not the implementation, but the idea of the algorithm and the correctness of termination condition.

- Learn about non-classical logics:
  Formalise a mutual exclusion solution in PLTL and check with the decision procedure whether it really is a solution (cp. for example [Eme90]).
- Compare logics:
  Formalise a certain block world situation in the three nonmonotonic logics circumscription, closed world assumption, and default logic. Compare the advantages and disadvantages. (See for example the sample session in section 8.11.)

8.10 ACCESS TO THE LWB

The binaries of version 1.0 is available for Sun Solaris vie the LWB home page. Forthcoming versions will probably be available for other systems, too.

The LWB can be used via a World Wide Web Browser, i.e. without previous installation of any part of the LWB. Just load the LWB home page (address: `http:// lwbwww.unibe.ch:8080/LWBinfo.html`) and choose the item `run a session`. Afterwards one can type the input in a window and submit it. A computer at the IAM in Bern calculates the results and sends them back. The number of simultaneous users and the CPU time per job is limited.

8.11 SAMPLE SESSION

In this appendix we present two sample sessions, one based on monotonic systems and the other on nonmonotonic systems. You can find larger sessions, especially where the programming language is used in part III of this thesis. In contrast to all other sessions in this thesis, the one in this section was inserted by copy-and-paste and not computed during typesetting.

```
   LWB  -  The Logics Workbench  1.0
   type 'help;' for help

> a := (~(p v q) & (s <-> r) -> q v p)
>      & (((d -> c) -> d) -> d);
```

We have a look at the properties of this ‘arbitrary’ formula in classical propositional logic (`cpc`).

```
> load(cpc);
   cpc user
cpc> b := simplify(a);
   b := (s <-> r) -> q v p
```

Now let us look at the formula `a` again. It is a conjunction, and the second conjunct has the form

```
cpc> a2 := a[2];
   a2 := ((d -> c) -> d) -> d
```

Now we compare a2 with the Peirce's formula ...

```
cpc> fml("peirce");
   ((p1 -> p2) -> p1) -> p1
```

...and realise that they are identical modulo renaming. Thus a2 must be provable in classical, but not in intuitionistic logic (ipc).

```
cpc> load(ipc);
   ipc cpc user
ipc> cpc::provable(a2);
   true
ipc> provable(a2);
   false
```

Now we embed a2 in the modal logic s4. The result must not be provable in s4; however, it is provable in s5.

```
ipc> load(s4, s5);
   s5 s4 ipc cpc user
ipc> ipc::s4(a2);
   box(box(box(box d -> box c) -> box d) -> box d)
s5> s4::provable(ipc::s4(a2));
   false
s5> s5::provable(ipc::s4(a2));
   true
```

The second example shows a typical difference between closed world assumption (cwa) and default logic (default). The theory t contains statements about the colours of a cube and a ball, namely: The ball or the cube is red, both ball and cube are either red or blue. We want to minimise the number of red objects. The closed world assumption theory of t restricted on red(ball) and red(cube) is inconsistent because it contains both ~red(ball) and ~red(cube).

```
> t := [red(ball) v red(cube),
>      red(ball) <-> ~blue(ball), red(cube) <-> ~blue(cube)];
load(cwa);
   cwa user
cwa> consistent(t, [red(ball), red(cube)]);
   false
cwa> cwa(t, [red(ball), red(cube)]);
   [red(ball) v red(cube),
    red(ball) <-> ~blue(ball), red(cube) <-> ~blue(cube),
    ~red(ball), ~red(cube)]
```

In default logic, we can express the minimality of red objects with the two rules in Delta. The first rule [true, ~red(ball), ~red(ball)] says: If we can add ~red(ball) to the theory without obtaining an inconsistent theory, then we add ~red(ball). The second rule states the same for ~red(cube).

We obtain two consistent extensions. One extension contains `~red(ball)` and the other one contains `~red(cube)`.

```
cwa> load(default);
   default cwa user
default> Delta := [ [true, ~red(ball), ~red(ball)],
default>            [true, ~red(cube), ~red(cube)] ];
default> extensions([t, Delta]);
   [ [red(ball) v red(cube),
      red(ball) <-> ~blue(ball), red(cube) <-> ~blue(cube),
      ~red(ball)],
     [red(ball) v red(cube),
      red(ball) <-> ~blue(ball), red(cube) <-> ~blue(cube),
      ~red(cube)] ]
```

8.12 SUMMARY

We have given an overview of the Logics Workbench (LWB), version 1.0 . See `http://lwbwww.unibe.ch:8080/LWBinfo.html` for more information.

9

TESTS

"How do they know the load limit on bridges, dad?" "They drive bigger and bigger trucks over the bridge until it breaks. Then they weigh the last truck and rebuild the bridge."

B. Watterson. Calvin and Hobbes.

9.1 INTRODUCTION

In this chapter we define tests for decision procedures for the logics K, KT, and S4.

All these tests are black box tests: We feed the decision procedure with a formula and check whether the output ('valid' resp. 'not valid') is correct. We assume that there is no possibility to see what is going on during the computation. This assumption is useful if we want to apply the tests to different sorts of decision procedures, but it is also the source of the so-called propagation problem: It could happen that the result is correct for a certain formula although something went wrong during the computation. Some experiments with the decision procedure for K in the LWB showed that the chances for this happening are good: We were able to build some mistakes into the procedure that influenced the result only very seldom. (Unfortunately the more sophisticated an algorithm is, the more possibilities there are for this sort of mistake.)

Therefore it is important to make tests with a large number of formulas. However, not only the number, but also the variety of these formulas matters, since otherwise the coverage can be poor. If for example $\leftrightarrow$ occurs in almost no formula, and the decision procedure does not convert $\leftrightarrow$ into $\rightarrow$ and $\wedge$, but has a special rule for $\leftrightarrow$, then it is probable that a mistake in this part of the procedure will not be detected.

First we collected formulas from the literature (textbooks about modal logic, but also papers on automated theorem proving) where the correct result is known. Even testing just a few formulas by hand is tedious. On the other hand, it is important to do these tests after changes, in order to detect unintentional side effects (read: mistakes). Thus an automatisation is mandatory. We did this with the help of a small program that takes the test files, applies the LWB to them, saves the output in a file, compares it with the expected output, and reports differences. Like this we can run all our tests with a simple command and just have to check afterwards whether any differences have been reported.

After a while it became clear that these lists were not sufficient, and we looked for other ways to generate test formulas. This was a time-consuming task and should not be repeated by every author of a decision procedure for K, KT, S4. Therefore we present tests for these three logics. Since the predominant number of formulas is generated by programs, the time-consuming and error prone typing is reduced to a minimum.

9.2 METHOD

The tests for K, KT, S4 consist of five parts:

1. Some simple formulas.
2. Ordering of standard formulas.
3. Random formulas.
4. Valid formulas.
5. Nonvalid formulas.

We discuss these five parts in the following subsections, giving both the LWB output (since there, no typing errors are possible) and a LaTeX table (which is more readable).

Note that the results in the parts 1., 2., 4. of our tests are the results of the LWB. We did not compare them with the results of another prover. If your prover returns another result, first check whether there is a typing error in the LaTeX table (by looking at the corresponding LWB session). Then check the result by hand, using for example one of the sequent calculi of chapter 5. If you have access to the LWB (for example via WWW, cp. chapter 8), then use `set("infolevel", 5);` to see what is going on.

9.2.1 Some simple formulas

In the first part we list 10 valid and 10 nonvalid formulas.

Of course this is on no account a sufficient test. However, theses formulas can be helpful while debugging in order to detect the obvious mistakes in a short time.

9.2.2 Ordering of standard formulas

Now we want to have a larger list of simple formulas. Let S be the set of standard formulas defined in figure 1.a in chapter 1. We list all implications $A \to C$ with $A \in S$ and $C \in S$ and $L \models A \to C$. Like this we obtain 1600 test formulas (some valid, some nonvalid) for K, KT, and S4.

9.2.3 Random formulas

First we define what 'random formula' means in this chapter.

Below, we define the procedure **`rnd_fml`** in the LWB programming language. Let rnd_fml be 'the corresponding mathematical function'. If A is the result of rnd_fml$(i, \{p_0, \ldots, p_n\}, j)$, where $i \in \mathbb{N}$, $j \in \{1, 2, \ldots\}$, then we know:

- $\text{vars}(A) \subseteq \{p_0, \ldots, p_n\}$
- $\text{depth}(A) \leq i$

The argument j is used as a starting point for the computation or the required random numbers. We will also make use of this function to generate the formulas in parts 4 and 5 of our tests.

The LWB function `rnd_fml` has three arguments. `i` is the upper limit for the depth of the generated random formula. If `i` is 0, or if a generated random number is equal to 0 modulo 8, then a leaf (generated with `rnd_leaf`) is returned. Otherwise this random number defines the main connective. The subformulas are generated with recursive calls of `rnd_fml`. The third argument of `rnd_fml` is a reference parameter, which is used for the generation of random numbers.

```
> proc : rnd_fml(i, vl, var j)
   #
  local j2;
   #
  begin
    if i = 0 then return rnd_leaf(vl, j);
    j2 := j;
    next_rnd(j);
    if j2 mod 8 = 0 then return rnd_leaf(vl, j);
    if j2 mod 8 = 1 then return rnd_fml(i-1, vl, j) &   rnd_fml(i-1, vl, j);
    if j2 mod 8 = 2 then return rnd_fml(i-1, vl, j) v   rnd_fml(i-1, vl, j);
    if j2 mod 8 = 3 then return rnd_fml(i-1, vl, j) ->  rnd_fml(i-1, vl, j);
    if j2 mod 8 = 4 then return rnd_fml(i-1, vl, j) <-> rnd_fml(i-1, vl, j);
    if j2 mod 8 = 5 then return ~ rnd_fml(i-1, vl, j);
    if j2 mod 8 = 6 then return box rnd_fml(i-1, vl, j);
    if j2 mod 8 = 7 then return dia rnd_fml(i-1, vl, j);
  end; # rnd_fml
```

`rnd_leaf` chooses a random element of the union of the list of variables `v` and `[true, false]`.

```
> proc : rnd_leaf(vl, var j)
  local j2;
  begin
    j2 := j;
    next_rnd(j);
    if j2 mod (nops(vl) + 2) = 0 then return true;
    if j2 mod (nops(vl) + 2) = 1 then return false;
    return vl[(j2 mod (nops(vl) + 2)) - 1];
  end; # rnd_leaf
```

`next_rnd` replaces the number `j` (an element of $\{1, 2, \ldots\}$) by a new random integer in $\{1, 2, \ldots\}$.

```
> proc : next_rnd(var j)
  begin
    j := (29 mult j) mod 65537;
  end; # next_rnd
```

In order to check the quality of `next_rnd`, we define the procedure `check`. `check(j)` returns the number of times the procedure `next_rnd` must be called until `j` has the start value again.

```
> proc : check(j)
  local start, counter;
  begin
    start := j;
    counter := 1;
    next_rnd(j);
    while ( j <> start ) do begin counter := counter + 1; next_rnd(j); end;
    print(counter);
  end; # check
```

The following call of check shows that all values in $\{1, \ldots, 65536\}$ are generated by `next_rnd` before a loop occurs.

```
> check(5234);
```

Here are some tests of `rnd_fml`. If you write your own `rnd_fml` you can use these values to check whether you have implemented the same function.

```
> j :== 276;
> for i := 1 to 10 do rnd_fml(3, [p0,p1,p2,p3], j);
> j :== 12333;
> for i := 1 to 10 do rnd_fml(4, [p0,p1], j);
```

For each logic we choose $9 \cdot 200$ of these random formulas and list the valid ones.

9.2.4 Valid formulas

We first compute a random formula A. Then we set $B :\equiv A \vee \neg A$. Obviously formula B is valid in K, KT, S4.

Using such formulas would result in a rather poor test, since many provers would immediately 'recognise' the structure of the formula. The LWB for example uses a two-sided sequent calculus, i.e. after applying (r$\vee$) and (r$\neg$) backwards, it obtains the sequent of the form $A, \Gamma \supset A, \Delta$, which is considered an axiom. The same problem could occur if we used the negation normal form of $\neg A$ instead of $\neg A$, although it would be already harder.

With the procedure `add_garbage` and by transforming one part of the formula into negation normal form we try to make it harder for the prover to 'see' the reason why this formula is valid.

```
> proc : add_garbage(a)
  local n, j;
    # We assume that a is in negation normal form.
    # The formula a -> add_garbage(a) is valid in K.
    proc add_garbage2(a, var n, var j)
    local b, c;
    begin
      b := a;
      if ( type(a) = BOX ) then b := box add_garbage2(a[1],n,j);
      if ( type(a) = DIA ) then b := dia add_garbage2(a[1],n,j);
      if ( type(a) = AND ) then b := add_garbage2(a[1],n,j) & add_garbage2(a[2],n,j);
      if ( type(a) = OR ) then b := add_garbage2(a[2],n,j) v add_garbage2(a[1],n,j);
      n := (n + 1) mod 10;
      if ( n = 5 ) then return b v rnd_fml(2, [p0,p3,p5,p7], j); else return b;
    end;
    #
  begin
    n := 3;
    j := 721;
    return add_garbage2(a, n, j);
  end; # add_garbage
```

```
> add_garbage(box p0 & dia(~p1 v p0) v dia p1);
> add_garbage(box box box box box box box box true);
> add_garbage(false & (p0 v ~ p1) & (p1 & p2) & (p2 v p3) & p4 & p5);
> add_garbage(false & (p0 v ~ p1) & (p1 & ~ ~ ~p2) & (p2 v p3) & p4);
> add_garbage(box(p0 v p1) & box(p0 v p1) & box(p0 v p1) & box(p0 v p1)
              & box(p0 v p1) & box(p0 v p1) & box(p0 v p1));
```

Let add_garbage be 'the corresponding mathematical functions'. We check for 3000 random formulas A whether add_garbage$(\mathrm{nnf}(\neg A) \vee A)$ is valid.

9.2.5 Nonvalid formulas

The origin of all these formulas are nonderivability results. For example it can be proved that $\mathsf{KT4Dum}^{\mathcal{H}} \nvdash \mathrm{Grz}$. The Hilbert-style calculus $\mathsf{KT4Dum}^{\mathcal{H}}$ is the calculus $\mathsf{K}^{\mathcal{H}}$ plus the formula schemes that correspond to the formulas T, 4, Dum as axioms. Thus no formula of the form $\bigwedge_{j \in \{1,\ldots,n\}}(\mathrm{Dum}\{A_j/p_0\}) \to \mathrm{Grz}$ can be valid, whatever formulas $A_1, \ldots, A_n$ we use to build the instances. (See chapter 11 for a thorough discussion of these issues.)

For each logic $L \in \{\mathsf{K}, \mathsf{KT}, \mathsf{S4}\}$ we choose three such nonderivability results, i.e. three pairs of formulas C, E such that $\mathsf{L}\mathsf{C}^{\mathcal{H}} \nvdash E$. For each pair C, E we compute 1000 tuples of random formulas $\langle A_1, A_2, A_3, A_4 \rangle$ and check whether $C\{A_1/p_0\} \wedge C\{A_2/p_0\} \wedge C\{A_3/p_0\} \wedge C\{A_4/p_0\} \to E$ is valid in L.

9.3 K

9.3.1 Some simple formulas

valid formulas	nonvalid formulas
$p_0 \to p_1 \lor p_0$	p_0
$\Box p_1 \land p_0 \to p_0 \land \Box p_1 \land \Box p_1$	$\Diamond\mathsf{true}$
$\Box(\Box p_2 \lor \Diamond(\neg p_2 \lor p_4))$	$p_1 \land \Box p_1 \lor \Box\Box\neg p_1$
$\Box p_0 \land \Box(p_0 \to p_1) \to \Box p_1$	$\Diamond\Box p_2 \lor \Diamond\Box\neg p_2$
$\Box p_0 \lor \Box p_1 \to \Box(p_0 \lor p_1)$	$\Box p_0 \to p_0$
$\Box(p_0 \land p_1) \leftrightarrow \Box p_0 \land \Box p_1$	$\neg\Box p_0 \to \Box\neg\Box p_0$
$\Box\neg p_0 \land \neg\Diamond p_1 \lor \Diamond p_0 \lor \Diamond p_1$	$\Box\neg p_1 \to \Box\neg\Diamond p_1$
$\Diamond p_0 \lor \Diamond p_1 \to \Diamond(p_0 \lor p_1)$	$\Diamond p_0 \land \Diamond p_1 \to \Diamond(p_0 \land p_1)$
$\Box\neg p_1 \lor \Diamond(((\Box p_2 \to \Diamond p_1) \to \Box p_2) \to \Box p_2)$	$\Diamond\Box\Diamond\Box p_0 \to \Diamond\Box p_0$
$\neg\Box(\Diamond p_0 \lor p_1) \to \Diamond\Box\neg p_0$	$\Diamond\Box p_0 \to \Diamond\Box\Diamond\Box p_0$

```
> k::provable(p0 -> p1 v p0);
> k::provable(box p1 & p0 -> p0 & box p1 & box p1);
> k::provable(box(box p2 v dia(~p2 v p4)));
> k::provable(box p0 & box(p0 -> p1) -> box p1);
> k::provable(box p0 v box p1 -> box(p0 v p1));
> k::provable(box(p0 & p1) <-> box p0 & box p1);
> k::provable(box ~p0 & ~dia p1 v dia p0 v dia p1);
> k::provable(dia p0 v dia p1 -> dia(p0 v p1));
> k::provable(box ~p1 v dia(((box p2 -> dia p1) -> box p2) -> box p2));
> k::provable(~box(dia p0 v p1) -> dia box ~p0);
> k::provable(p0);
> k::provable(dia true);
> k::provable(p1 & box p1 v box box ~p1);
> k::provable(dia box p2 v dia box ~p2);
> k::provable(box p0 -> p0);
> k::provable(~box p0 -> box ~box p0);
> k::provable(box ~p1 -> box ~dia p1);
> k::provable(dia p0 & dia p1 -> dia(p0 & p1));
> k::provable(dia box dia box p0 -> dia box p0);
> k::provable(dia box p0 -> dia box dia box p0);
```

9.3.2 Ordering of standard formulas

For $A, B \in S$ we have $\mathsf{K} \models A \to B$ iff there is an entry K in figure 9.a or figure 9.b on line A and column B. Example: $\mathsf{K} \models \mathrm{Pt} \to \mathrm{M}$, but $\mathsf{K} \not\models \mathrm{M} \to \mathrm{Pt}$.

Obviously $\mathsf{K} \models A \to A$ for all $A \in S$. 75 of the 1600 formulas are valid in K, 1525 are not valid in K.

The corresponding LWB session:

```
> D :== box p -> dia p;
  D2 :== dia true;
  T :== box p -> p;
```

	D	D_2	T	4	4_M	5	5_M	B	B_M	G	G_0	H	H^+	L	L^+	L^{++}	M	M_2	M_3	Pt
D	K	K	KT	S4	S4															
D_2	K	K	KT	S4	S4															
T	KT	KT	K	S4	S4															
4	KT	KT	KT	K	K															
4_M	KT	KT	KT	S4	K															
5	KT	KT	KT	S4	S4	K	K	KT	KT	KT										
5_M	KT	KT	KT	S4	S4		K		KT											
B	KT	KT	KT	S4	S4			K	K											
B_M	KT	KT	KT	S4	S4				K											
G	KT	KT	KT	S4	S4					K										
G_0	KT	KT	KT	S4	S4						K									
H	KT	KT	KT	S4	S4							K	KT							
H^+	KT	KT	KT	S4	S4							K	K							
L	KT	KT	KT	S4	S4							K	KT	K	KT	S4				
L^+	KT	KT	KT	S4	S4							K	K	K	K	S4				
L^{++}	KT	KT	KT	S4	S4							KT	KT	KT	KT	K				
M	K	K	KT	S4	S4												K	S4	S4	KT
M_2	K	K	KT	S4	S4												S4	K	S4	S4
M_3	K	K	KT	S4	S4														K	
Pt	K	K	KT	S4	S4												K	S4	S4	K
W	KT	KT	KT	S4	S4	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	S4	KT	S4	KT	KT
W_0	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT
Z	KT	KT	KT	S4	S4					KT		KT	KT	KT	KT	S4				
Dum	KT	KT	KT	S4	S4															
Dum_1	KT	KT	KT	S4	S4															
Dum_2	KT	KT	KT	S4	S4															
Dum_3	KT	KT	KT	S4	S4															
Dum_4	KT	KT	KT	S4	S4															
Grz	KT	KT	K	S4	S4															
Grz_1	KT	KT	KT	S4	S4												S4	S4	S4	S4
Grz_2	KT	KT	KT	S4	S4															
Grz_3	KT	KT	KT	S4	S4												S4	S4	S4	S4
Grz_4	KT	KT	KT	S4	S4															
Grz_5	KT	KT	KT	S4	S4														S4	
F	KT	KT	KT	S4	S4															
H_s	KT	KT	KT	S4	S4															
P	KT	KT	KT	S4	S4							KT	KT							
R	KT	KT	KT	S4	S4							KT	KT							
X	KT	KT	KT	S4	S4															
Zem	KT	KT	KT	S4	S4															

Figure 9.a: Ordering of standard formulas. Part 1.

	W	W_0	Z	Dum	Dum_1	Dum_2	Dum_3	Dum_4	Grz	Grz_1	Grz_2	Grz_3	Grz_4	Grz_5	F	H_s	P	R	X	Zem
D																			KT	
D_2																			KT	
T																			KT	
4																			KT	
4_M																			KT	
5																			KT	
5_M																			KT	
B																			KT	
B_M																			KT	
G																			KT	
G_0																			KT	
H																			KT	
H^+																			KT	
L																			KT	
L^+																			KT	
L^{++}																			KT	
M																			KT	
M_2																			KT	
M_3																			KT	
Pt																			KT	
W	K		K	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT
W_0	KT	K	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	KT	K
Z			K	KT	KT	KT	KT	KT							KT			KT	KT	KT
Dum				K		KT		K											KT	
Dum_1				KT	K	KT	KT	K											KT	
Dum_2				KT		K		KT											KT	
Dum_3				KT	KT	KT	K	KT											K	
Dum_4				KT		KT		K											KT	
Grz				K		KT		K	K		KT		K						KT	
Grz_1				KT	K	KT	KT	K	KT	K	KT	KT	K						KT	
Grz_2				KT		K		KT	KT		K		KT						KT	
Grz_3				KT	KT	KT	K	KT	KT	KT	KT	K	KT						K	
Grz_4				KT		KT		K	KT		KT		K						KT	
Grz_5														K					KT	
F															K				KT	
H_s																K			KT	KT
P																	K	KT	KT	KT
R																		K	KT	KT
X																			K	
Zem																			KT	K

Figure 9.b: Ordering of standard formulas. Part 2.

```
  A4 :== box p -> box box p;
  A4m :== box p & dia q -> dia(box p & q);
  A5 :== dia p -> box dia p;
  A5m :== dia p & dia q -> dia(dia p & q);
  B :== p -> box dia p;
  Bm :== p & dia q -> dia(dia p & q);
   #
  G :== dia box p -> box dia p;
  G0 :== dia(p & box q) -> box(p v dia q);
  H :== box(p v q) & box(box p v q) & box(p v box q) -> box p v box q;
  Hp :== box(box p v q) & box(p v box q) -> box p v box q;
  L :== box(p & box p -> q) v box(q & box q -> p);
  Lp :== box(box p -> q) v box(box q -> p);
  Lpp :== box(box p -> box q) v box(box q -> box p);
   #
  M :== box dia p -> dia box p;
  M2 :== dia box(p -> box p);
  M3 :== box dia p & box dia q -> dia(p & q);
  Pt :== box(p v dia p) -> dia(p & box p);
   #
  W :== box(box p -> p) -> box p;
  W0 :== box dia true -> box false;
  Z :== box(box p -> p) -> (dia box p -> box p);
   #
  Dum :== box(box(p -> box p) -> p) -> (dia box p -> p);
  Dum1 :== box(box(p -> box p) -> p) -> (dia box p -> box p);
  Dum2 :== box(box(p -> box p) -> box p) -> (dia box p -> p);
  Dum3 :== box(box(p -> box p) -> box p) -> (dia box p -> box p);
  Dum4 :== box(box(p -> box p) -> p)     -> (dia box p -> p v box p);
  Grz :== box(box(p -> box p) -> p) -> p;
  Grz1 :== box(box(p -> box p) -> p) -> box p;
  Grz2 :== box(box(p -> box p) -> box p) -> p;
  Grz3 :== box(box(p -> box p) -> box p) -> box p;
  Grz4 :== box(box(p -> box p) -> p)     -> p v box p;
  Grz5 :== box(box(p -> box q) -> box q) & box(box(~p -> box q) -> box q) -> box q;
   #
  F :== (dia box p -> q) v box(box q -> p);
  Hs :== p -> box(dia p -> p);
  P :== dia box dia p -> (p -> box p);
  R :== dia box p -> (p -> box p);
  X :== box box p -> box p;
  Zem :== box dia box p -> (p -> box p);
> l :== [D,    D2,   T,    A4,   A4m,  A5,   A5m,  B,    Bm,  G,
         G0,   H,    Hp,   L,    Lp,   Lpp,  M,    M2,   M3,  Pt,
         W,    W0,   Z,    Dum,  Dum1, Dum2, Dum3, Dum4, Grz, Grz1,
         Grz2, Grz3, Grz4, Grz5, F,    Hs,   P,    R,    X,   Zem];
  ln :== ["D",    "D2",   "T",    "A4",   "A4m",  "A5",   "A5m",  "B",    "Bm",  "G",
          "G0",   "H",    "Hp",   "L",    "Lp",   "Lpp",  "M",    "M2",   "M3",  "Pt",
          "W",    "W0",   "Z",    "Dum",  "Dum1", "Dum2", "Dum3", "Dum4", "Grz", "Grz1",
          "Grz2", "Grz3", "Grz4", "Grz5", "F",    "Hs",   "P",    "R",    "X",   "Zem"];
```

The first table:

```
> k_n :== 0; kt_n :== 0; s4_n :== 0;
> for i := 1 to 40 do
  begin
    line :== [];
    for j := 1 to 20 do
    begin
      if k::provable(l[i] -> l[j])
      then begin
        k_n :== k_n + 1;
        line :== concat(line, [[ln[j], "K"]]);
      end;
      else if kt::provable(l[i] -> l[j])
      then begin
        kt_n :== kt_n + 1;
        line :== concat(line, [[ln[j], "KT"]]);
      end;
      else if s4::provable(l[i] -> l[j])
      then begin
        s4_n :== s4_n + 1;
        line :== concat(line, [[ln[j], "S4"]]);
      end;
    end;
    print(ln[i], ": ", line);
  end;
```

The second table:

```
> for i := 1 to 40 do
  begin
    line :== [];
    for j := 21 to 40 do
    begin
      if k::provable(l[i] -> l[j])
      then begin
        k_n :== k_n + 1;
        line :== concat(line, [[ln[j], "K"]]);
      end;
      else if kt::provable(l[i] -> l[j])
      then begin
        kt_n :== kt_n + 1;
        line :== concat(line, [[ln[j], "KT"]]);
      end;
      else if s4::provable(l[i] -> l[j])
      then begin
        s4_n :== s4_n + 1;
        line :== concat(line, [[ln[j], "S4"]]);
      end;
    end;
    print(ln[i], ": ", line);
  end;
```

Finally the number of formulas valid in K, KT, and S4:

i	j
2	2, 3, 8, 10, 20, 26, 32, 37, 56, 58, 69, 74, 75, 80, 81, 85, 86, 93, 98, 99, 104, 106, 110, 115, 116, 117, 122, 123, 124, 128, 130, 133, 134, 138, 140, 141, 152, 154, 170, 171, 176, 181, 186, 194, 195, 196, 200
3	2, 10, 18, 24, 27, 30, 34, 43, 48, 50, 58, 59, 61, 66, 72, 74, 77, 78, 90, 96, 101, 106, 115, 120, 122, 130, 133, 134, 144, 163, 168, 171, 173, 186, 192, 194, 198
4	2, 10, 11, 16, 18, 19, 26, 29, 40, 50, 58, 64, 66, 67, 70, 78, 86, 88, 90, 112, 114, 115, 118, 122, 133, 136, 147, 157, 158, 160, 163, 165, 170, 173, 174, 178, 179, 181, 184, 186, 187, 195, 197, 198
5	5, 6, 8, 11, 26, 29, 32, 34, 42, 56, 58, 62, 77, 80, 82, 89, 101, 104, 115, 118, 128, 152, 155, 170, 176, 181, 194, 200
6	2, 5, 19, 24, 34, 48, 49, 50, 58, 72, 74, 96, 102, 109, 120, 122, 130, 131, 138, 144, 150, 162, 168, 178, 179, 187, 192, 193
7	3, 5, 10, 14, 16, 18, 26, 28, 34, 40, 42, 43, 64, 82, 83, 86, 88, 107, 112, 115, 126, 136, 138, 141, 142, 154, 155, 160, 170, 174, 181, 184
8	8, 11, 19, 21, 26, 29, 32, 50, 56, 61, 74, 80, 104, 106, 122, 125, 128, 147, 152, 155, 158, 165, 176, 186, 187, 194, 200
9	2, 24, 34, 42, 48, 50, 72, 90, 94, 96, 114, 117, 120, 125, 144, 147, 162, 168, 192
10	10, 13, 16, 27, 34, 40, 50, 64, 66, 82, 83, 88, 91, 101, 112, 136, 138, 139, 147, 150, 154, 160, 170, 182, 184, 189, 190, 194

Figure 9.c: Random formulas valid in K.

```
> k_n :== 0; kt_n :== 0; s4_n :== 0;
> for i := 1 to 40 do for j := 1 to 40 do if k::provable(l[j]->l[i]) then k_n :== k_n + 1;
  print(k_n);
> for i := 1 to 40 do for j := 1 to 40 do if kt::provable(l[j]->l[i]) then kt_n :== kt_n + 1;
  print(kt_n);
> for i := 1 to 40 do for j := 1 to 40 do if s4::provable(l[j]->l[i]) then s4_n :== s4_n + 1;
  print(s4_n);
```

9.3.3 Random formulas

For $i \in \{2, 3, \ldots 10\}$, $j \in \{1, \ldots, 200\}$ we have $\mathsf{K} \models \text{rnd\_fml}(i, \{p_0, p_1, p_2, p_3\}, 200i + j)$ iff there is an entry j on line i in figure 9.c. Example: $\mathsf{K} \models \text{rnd\_fml}(3, \{p_0, p_1, p_2, p_3\}, 618)$, $\mathsf{K} \not\models \text{rnd\_fml}(3, \{p_0, p_1, p_2, p_3\}, 619)$, $\mathsf{K} \models \text{rnd\_fml}(3, \{p_0, p_1, p_2, p_3\}, 627)$.

290 of the 1800 formulas are valid in K, 1510 are not valid in K.

The corresponding LWB session:

```
> load(k);
> proc : k_random()
  local i, j, j2, l, a, total;
  begin
    total := 0;
    for i := 2 to 10 do
    begin
      l :== [];
      for j := 1 to 200 do
      begin
```

```
        j2 :== (i mult 200) + j;
        a :== rnd_fml(i, [p0,p1,p2,p3], j2);
        if k::provable(a) then append(l, j);
      end;
      print(i, " : ", l);
      total := total + nops(l);
    end;
    print("total: ", total, " valid and ", 9 mult 200 - total, " nonvalid formulas");
  end; # k_random
> k_random();
```

9.3.4 Valid formulas

For all $d \in \{3, 5, 7\}$ we compute 1000 random formulas A with depth d, variables in $\{p_0, \dots, p_6\}$ and initial value 117 for j, and check whether the formula add_garbage(nnf($\neg A$)) $\vee A$ is valid in K. For each d three formulas are printed in order to show that they are in general nontrivial.

Attention: In the following LWB session we use only 10 random formulas instead of 1000, since otherwise the typesetting process would take too much time.

```
> j :== 117;
  for i := 1 to 1000
  do begin
    a :== rnd_fml(3, [p0,p1,p2,p3,p4,p5,p6], j);
    b :== add_garbage(nnf(~a));
    if ( not k::provable(b v a) ) then print("***** error *****");
    if ( ( i = 17 ) or ( i = 361 ) or ( i = 752 ) ) then print(b v a);
  end;
  print("done");
> j :== 117;
  for i := 1 to 1000
  do begin
    a :== rnd_fml(5, [p0,p1,p2,p3,p4,p5,p6], j);
    b :== add_garbage(nnf(~a));
    if ( not k::provable(b v a) ) then print("***** error *****");
    if ( ( i = 17 ) or ( i = 361 ) or ( i = 752 ) ) then print(b v a);
  end;
  print("done");
> j :== 117;
  for i := 1 to 10 # Attention: Use 1000 instead of 10 for your tests.
  do begin
    a :== rnd_fml(7, [p0,p1,p2,p3,p4,p5,p6], j);
    b :== add_garbage(nnf(~a));
    if ( not k::provable(b v a) ) then print("***** error *****");
    if ( ( i = 3 ) or ( i = 7 ) or ( i = 8 ) ) then print(b v a);
  end;
  print("done");
```

9.3.5 Nonvalid formulas

We choose the nonderivability results $\mathsf{KT}^{\mathcal{H}} \nvdash 4$, $\mathsf{K4}^{\mathcal{H}} \nvdash 5$, and $\mathsf{KDum}^{\mathcal{H}} \nvdash$ Grz. The 1000 random formulas with depth 3, variables in $\{p_0, p_1, p_2, p_3, p_4\}$, and initial value 234. See the following LWB session for details.

```
> for i := 1 to 1000
  do begin
    tuple :== [];
    j :== 234;
    a1 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a2 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a3 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a4 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    b :== T{a1/p0} & T{a2/p0} & T{a3/p0} & T{a4/p0} -> A4;
    if ( i = 123 ) then print(b);
    if ( k::provable(b) ) then print("***** error *****");
  end;
  print("done");
> for i := 1 to 1000
  do begin
    tuple :== [];
    j :== 234;
    a1 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a2 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a3 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a4 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    b :== A4{a1/p0} & A4{a2/p0} & A4{a3/p0} & A4{a4/p0} -> A5;
    if ( i = 123 ) then print(b);
    if ( k::provable(b) ) then print("***** error *****");
  end;
  print("done");
> for i := 1 to 1000
  do begin
    tuple :== [];
    j :== 234;
    a1 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a2 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a3 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a4 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    b :== Dum{a1/p0} & Dum{a2/p0} & Dum{a3/p0} & Dum{a4/p0} -> Grz;
    if ( i = 123 ) then print(b);
    if ( k::provable(b) ) then print("***** error *****");
  end;
  print("done");
```

9.4 KT

9.4.1 Some simple formulas

valid formulas	nonvalid formulas
$p_0 \to p_1 \lor p_0$	p_0
$\Diamond(p_1 \leftrightarrow p_1)$	$\Diamond\Box\neg p_0$
$\Box p_0 \to p_0$	$p_1 \land \Box p_1 \lor \Box\Box\neg p_1$
$\Box p_0 \land \neg\Diamond p_1) \to p_0 \land \neg p_1$	$\Diamond\Box p_2 \lor \Diamond\Box\neg p_2$
$\Box p_1 \land p_0 \to p_0 \land \Box p_1 \land \Box p_1$	$\neg\Box p_0 \to \Box\neg\Box p_0$
$\Box(\Box p_2 \lor \Diamond(\neg p_2 \lor p_4))$	$\Box\neg p_1 \to \Box\neg\Diamond p_1$
$\Box(p_0 \land p_1) \leftrightarrow \Box p_0 \land \Box p_1$	$\Diamond p_0 \land \Diamond p_1 \to \Diamond(p_0 \land p_1)$
$\Diamond p_0 \lor \Diamond p_1 \to \Diamond(p_0 \lor p_1)$	$\Diamond\Box\Diamond\Box p_0 \to \Diamond\Box p_0$
$\Diamond(((\Box p_2 \to \Diamond p_1) \to \Box p_2) \to \Box p_2)$	$\Diamond\Box p_0 \to \Diamond\Box\Diamond\Box p_0$
$\neg\Box(\Diamond p_0 \lor p_1) \to \Diamond\Diamond\Diamond\neg p_0$	$\Box(p_2 \land p_0 \land p_1) \land \Box(p_0 \land p_1) \to \Box p_0 \land \Box(\Box p_1 \land p_2)$

```
> kt::provable(p0 -> p1 v p0);
> kt::provable(dia(p1 <-> p1));
> kt::provable(box p0 -> p0);
> kt::provable((box p0 & ~dia p1) -> p0 & ~p1);
> kt::provable(box p1 & p0 -> p0 & box p1 & box p1);
> kt::provable(box(box p2 v dia(~p2 v p4)));
> kt::provable(box(p0 & p1) <-> box p0 & box p1);
> kt::provable(dia p0 v dia p1 -> dia(p0 v p1));
> kt::provable(dia(((box p2 -> dia p1) -> box p2) -> box p2));
> kt::provable(~box(dia p0 v p1) -> dia dia dia ~p0);
> kt::provable(p0);
> kt::provable(dia box ~p0);
> kt::provable(p1 & box p1 v box box ~p1);
> kt::provable(dia box p2 v dia box ~p2);
> kt::provable(~box p0 -> box ~box p0);
> kt::provable(box ~p1 -> box ~dia p1);
> kt::provable(dia p0 & dia p1 -> dia(p0 & p1));
> kt::provable(dia box dia box p0 -> dia box p0);
> kt::provable(dia box p0 -> dia box dia box p0);
> kt::provable(box(p2 & p0 & p1) & box(p0 & p1) -> box p0 & box(box p1 & p2));
```

9.4.2 Ordering of standard formulas

For $A, B \in S$ we have $\mathsf{KT} \models A \to B$ iff there is an entry K or KT in figure 9.a or figure 9.b on line A and column B. Example: $\mathsf{KT} \vdash \mathrm{Dum}_1 \to \mathrm{Dum}$, $\mathsf{KT} \vdash \mathrm{Dum}_1 \to \mathrm{Dum}_1$, but $\mathsf{KT} \nvdash \mathrm{Dum} \to \mathrm{Dum}_1$.

Obviously $\mathsf{KT} \vdash A \to A$, $\mathsf{KT} \vdash A \to \mathrm{D}$, $\mathsf{KT} \vdash A \to \mathrm{D}_2$, $\mathsf{KT} \vdash A \to \mathrm{T}$, $\mathsf{KT} \vdash A \to \mathrm{X}$, $\mathsf{KT} \vdash \mathrm{W}_0 \to A$ for all $A \in S$. 350 of the 1600 formulas are valid in KT, 1250 are not valid in KT. See section 9.3.2 for the corresponding LWB session.

9.4.3 Random formulas

For $i \in \{2, 3, \ldots 10\}$, $j \in \{1, \ldots, 200\}$ we have $\mathsf{KT} \models \mathrm{rnd\_fml}(i, \{p_0, p_1, p_2, p_3\}, 200i + j)$ iff there is an entry j on line i in figure 9.d. Example: $\mathsf{KT} \models \mathrm{rnd\_fml}(3, \{p_0, p_1, p_2, p_3\}, 618)$, $\mathsf{KT} \not\models \mathrm{rnd\_fml}(3, \{p_0, p_1, p_2, p_3\}, 619)$, $\mathsf{KT} \models \mathrm{rnd\_fml}(3, \{p_0, p_1, p_2, p_3\}, 627)$.

435 of the 1800 formulas are valid in KT, 1365 are not valid in KT.

The corresponding LWB session:

```
> load(kt);
> proc : kt_random()
```

i	j
2	2, 3, 8, 10, 20, 26, 32, 37, 39, 56, 58, 69, 74, 75, 79, 80, 81, 85, 86, 87, 93, 98, 99, 104, 106, 110, 111, 115, 116, 117, 122, 123, 124, 127, 128, 130, 133, 134, 135, 138, 140, 141, 152, 154, 167, 170, 171, 176, 181, 186, 191, 194, 195, 196, 200
3	2, 10, 18, 24, 27, 30, 34, 43, 47, 48, 50, 53, 55, 58, 59, 61, 66, 72, 74, 77, 78, 82, 90, 95, 96, 101, 106, 111, 115, 120, 122, 130, 133, 134, 143, 144, 162, 163, 167, 168, 171, 173, 178, 183, 186, 191, 192, 194, 198, 199
4	2, 7, 10, 11, 16, 18, 19, 23, 26, 29, 35, 39, 40, 42, 47, 50, 55, 58, 64, 66, 67, 69, 70, 78, 86, 88, 90, 103, 112, 114, 115, 118, 119, 122, 130, 133, 136, 147, 151, 157, 158, 160, 163, 165, 167, 170, 173, 174, 178, 179, 181, 184, 186, 187, 195, 197, 198
5	1, 5, 6, 7, 8, 11, 18, 26, 29, 32, 34, 42, 56, 58, 62, 66, 74, 77, 80, 82, 87, 89, 90, 95, 101, 104, 111, 114, 115, 118, 127, 128, 130, 138, 143, 152, 155, 159, 163, 167, 170, 174, 176, 181, 183, 194, 200
6	2, 5, 14, 15, 19, 24, 31, 34, 37, 38, 48, 49, 50, 58, 62, 69, 72, 74, 75, 79, 81, 87, 90, 96, 98, 102, 103, 109, 120, 122, 130, 131, 138, 144, 150, 151, 154, 159, 162, 168, 170, 178, 179, 183, 187, 192, 193, 199
7	3, 5, 10, 14, 15, 16, 18, 23, 26, 28, 31, 34, 35, 40, 42, 43, 63, 64, 71, 75, 79, 82, 83, 86, 87, 88, 102, 107, 108, 112, 115, 126, 127, 129, 135, 136, 138, 141, 142, 146, 150, 151, 154, 155, 160, 166, 170, 174, 181, 184, 190, 191
8	7, 8, 11, 14, 19, 21, 26, 29, 32, 38, 39, 50, 53, 56, 61, 63, 74, 79, 80, 87, 91, 104, 106, 122, 125, 128, 130, 133, 147, 151, 152, 154, 155, 158, 165, 167, 173, 175, 176, 186, 187, 194, 199, 200
9	2, 7, 23, 24, 29, 30, 31, 34, 39, 42, 46, 47, 48, 50, 61, 62, 72, 74, 90, 94, 96, 106, 114, 117, 120, 125, 127, 130, 134, 135, 138, 144, 147, 162, 165, 168, 178, 191, 192
10	2, 10, 13, 16, 21, 27, 34, 40, 50, 64, 66, 79, 82, 83, 88, 90, 91, 92, 98, 101, 103, 111, 112, 118, 130, 131, 134, 136, 138, 139, 143, 147, 150, 154, 160, 170, 174, 175, 182, 184, 189, 190, 194

Figure 9.d: Random formulas valid in KT.

```
  local i, j, j2, l, a, total;
  begin
    total := 0;
    for i := 2 to 10 do
    begin
      l :== [];
      for j := 1 to 200 do
      begin
        j2 :== (i mult 200) + j;
        a :== rnd_fml(i, [p0,p1,p2,p3], j2);
        if kt::provable(a) then append(l, j);
      end;
      print(i, " : ", l);
      total := total + nops(l);
    end;
    print("total: ", total, " valid and ", 9 mult 200 - total, " nonvalid formulas");
  end; # kt_random
> kt_random();
```

9.4.4 Valid formulas

For all $d \in \{3, 5, 7\}$ we compute 1000 random formulas A with depth d, variables in $\{p_0, \dots, p_6\}$ and initial value 117 for j, and check whether the formula add_garbage(nnf($\neg A$)) $\vee$ A is valid in KT.

Attention: In the following LWB session we use only 10 random formulas instead of 1000, since otherwise the typesetting process would take too much time.

```
> j :== 117;
  for i := 1 to 1000
  do begin
    a :== rnd_fml(3, [p0,p1,p2,p3,p4,p5,p6], j);
    b :== add_garbage(nnf(~a));
    if ( not kt::provable(b v a) ) then print("***** error *****");
  end;
  print("done");
> j :== 117;
  for i := 1 to 1000
  do begin
    a :== rnd_fml(5, [p0,p1,p2,p3,p4,p5,p6], j);
    b :== add_garbage(nnf(~a));
    if ( not kt::provable(b v a) ) then print("***** error *****");
  end;
  print("done");
> j :== 117;
  for i := 1 to 10 # Attention: Use 1000 instead of 10 for your tests.
  do begin
    a :== rnd_fml(7, [p0,p1,p2,p3,p4,p5,p6], j);
    b :== add_garbage(nnf(~a));
    if ( not kt::provable(b v a) ) then print("***** error *****");
  end;
  print("done");
```

9.4.5 Nonvalid formulas

We choose the nonderivability results $\mathsf{KT4}^{\mathcal{H}} \nvdash$ Dum, $\mathsf{KTB}^{\mathcal{H}} \nvdash 5$, and $\mathsf{KTDum}^{\mathcal{H}} \nvdash$ Grz. The 1000 random formulas with depth 3, variables in $\{p_0, p_1, p_2, p_3, p_4\}$, and initial value 234.

```
> for i := 1 to 1000
  do begin
    tuple :== [];
    j :== 234;
    a1 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a2 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a3 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a4 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    b :== A4{a1/p0} & A4{a2/p0} & A4{a3/p0} & A4{a4/p0} -> Dum;
    if ( i = 123 ) then print(b);
    if ( kt::provable(b) ) then print("***** error *****");
  end;
  print("done");
> for i := 1 to 1000
  do begin
    tuple :== [];
    j :== 234;
    a1 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a2 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a3 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a4 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    b :== B{a1/p0} & B{a2/p0} & B{a3/p0} & B{a4/p0} -> A5;
    if ( i = 123 ) then print(b);
    if ( kt::provable(b) ) then print("***** error *****");
  end;
  print("done");
> for i := 1 to 1000
  do begin
    tuple :== [];
    j :== 234;
    a1 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a2 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a3 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a4 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    b :== Dum{a1/p0} & Dum{a2/p0} & Dum{a3/p0} & Dum{a4/p0} -> Grz;
    if ( i = 123 ) then print(b);
    if ( kt::provable(b) ) then print("***** error *****");
  end;
  print("done");
```

9.5 S4

9.5.1 Some simple formulas

valid formulas	nonvalid formulas
$p_0 \to p_1 \lor p_0$	p_0
$\Diamond(p_1 \leftrightarrow p_1)$	$\Diamond\Box\neg p_0$
$\Box p_0 \to p_0$	$\Diamond(p_1 \land p_2) \lor \Diamond\Diamond\neg p_1$
$\Box p_0 \land \neg\Diamond p_1) \to p_0 \land \neg p_1$	$p_1 \land \Box p_1 \lor \Box\Box\neg p_1$
$\Box(\Box p_2 \lor \Diamond(\neg p_2 \lor p_4))$	$\Diamond\Box p_2 \lor \Diamond\Box\neg p_2$
$\Box(p_0 \land p_1) \leftrightarrow \Box p_0 \land \Box p_1$	$\neg\Box p_0 \to \Box\neg\Box p_0$
$\Diamond p_0 \lor \Diamond p_1 \to \Diamond(p_0 \lor p_1)$	$\Diamond p_0 \land \Diamond p_1 \to \Diamond(p_0 \land p_1)$
$\neg\Diamond(\Diamond p_0 \lor p_1) \to \Box\Box\neg p_0$	$\Box(p_0 \lor p_1) \to \Box p_0 \lor \Box p_1$
$\Diamond\Box\Diamond\Box p_0 \to \Diamond\Box p_0$	$\Diamond p_1 \land \Box p_2 \to \Diamond(p_1 \land p_3) \lor \Box p_1$
$\Diamond\Box p_0 \to \Diamond\Box\Diamond\Box p_0$	$\Box(p_0 \lor p_1) \to \Box p_0 \lor \Box p_1$

```
> s4::provable(p0 -> p1 v p0);
> s4::provable(dia(p1 <-> p1));
> s4::provable(box p0 -> p0);
> s4::provable((box p0 & ~dia p1) -> p0 & ~p1);
> s4::provable(box(box p2 v dia(~p2 v p4)));
> s4::provable(box(p0 & p1) <-> box p0 & box p1);
> s4::provable(dia p0 v dia p1 -> dia(p0 v p1));
> s4::provable(~dia(dia p0 v p1) -> box box ~p0);
> s4::provable(dia box dia box p0 -> dia box p0);
> s4::provable(dia box p0 -> dia box dia box p0);
> s4::provable(p0);
> s4::provable(dia box ~p0);
> s4::provable(dia(p1 & p2) v dia dia ~p1);
> s4::provable(p1 & box p1 v box box ~p1);
> s4::provable(dia box p2 v dia box ~p2);
> s4::provable(~box p0 -> box ~box p0);
> s4::provable(dia p0 & dia p1 -> dia(p0 & p1));
> s4::provable(box(p0 v p1) -> box p0 v box p1);
> s4::provable(dia p1 & box p2 -> dia(p1 & p3) v box p1);
> s4::provable(box(p0 v p1) -> box p0 v box p1);
```

9.5.2 Ordering of standard formulas

For $A, B \in S$ we have S4 $\models A \to B$ iff there is an entry K or KT or S4 in figure 9.a or figure 9.b on line A and column B. Example: S4 $\models \mathrm{H} \to 4_{\mathrm{M}}$, S4 $\models \mathrm{H} \to \mathrm{T}$, but S4 $\not\models 4_{\mathrm{M}} \to H$.

Obviously S4 $\models A \to A$, S4 $\models A \to \mathrm{D}$, S4 $\models A \to \mathrm{D}_2$, S4 $\models A \to \mathrm{T}$, S4 $\models A \to 4$, S4 $\models A \to 4_{\mathrm{M}}$, S4 $\models A \to \mathrm{X}$, S4 $\models \mathrm{W}_0 \to A$ for all $A \in S$. Also S4 $\models \mathrm{W}_0 \to A$ for all $A \in S$. 446 of the 1600 formulas are valid in S4, 1154 are not valid in S4.

See section 9.3.2 for the corresponding LWB session.

9.5.3 Random formulas

For $i \in \{2, 3, \ldots 10\}$, $j \in \{1, \ldots, 200\}$ we have S4 $\models \mathrm{rnd\_fml}(i, \{p_0, p_1, p_2, p_3\}, 200i + j)$ iff there is an entry j on line i in figure 9.e. Example: S4 $\models \mathrm{rnd\_fml}(3, \{p_0, p_1, p_2, p_3\}, 618)$, S4 $\not\models \mathrm{rnd\_fml}(3, \{p_0, p_1, p_2, p_3\}, 619)$, S4 $\not\models \mathrm{rnd\_fml}(3, \{p_0, p_1, p_2, p_3\}, 630)$.

441 of the 1800 formulas are valid in S4, 1359 are not valid in S4.

The corresponding LWB session:

```
> load(s4);
```

i	j
2	2, 3, 8, 10, 20, 26, 32, 37, 39, 56, 58, 69, 74, 75, 79, 80, 81, 85, 86, 87, 93, 98, 99, 104, 106, 110, 111, 115, 116, 117, 122, 123, 124, 127, 128, 130, 133, 134, 135, 138, 140, 141, 152, 154, 167, 170, 171, 176, 181, 186, 191, 194, 195, 196, 200
3	2, 10, 18, 24, 27, 30, 34, 43, 47, 48, 50, 53, 55, 58, 59, 61, 66, 72, 74, 77, 78, 82, 90, 95, 96, 101, 106, 111, 115, 120, 122, 130, 133, 134, 143, 144, 162, 163, 167, 168, 171, 173, 178, 183, 186, 191, 192, 194, 198, 199
4	2, 7, 10, 11, 16, 18, 19, 23, 26, 29, 35, 39, 40, 42, 47, 50, 55, 58, 64, 66, 67, 69, 70, 78, 86, 88, 90, 103, 112, 114, 115, 118, 119, 122, 130, 133, 136, 147, 151, 157, 158, 160, 163, 165, 167, 170, 173, 174, 178, 179, 181, 184, 186, 187, 195, 197, 198
5	1, 5, 6, 7, 8, 11, 18, 26, 29, 32, 34, 42, 56, 58, 62, 66, 74, 77, 80, 82, 87, 89, 90, 95, 101, 104, 111, 114, 115, 118, 127, 128, 130, 138, 143, 152, 155, 159, 163, 167, 170, 174, 176, 181, 183, 194, 200
6	2, 5, 14, 15, 19, 24, 31, 34, 37, 38, 48, 49, 50, 58, 62, 69, 72, 74, 75, 79, 81, 87, 90, 96, 98, 102, 103, 109, 120, 122, 130, 131, 138, 144, 150, 151, 154, 159, 162, 168, 170, 178, 179, 183, 187, 192, 193, 199
7	3, 5, 10, 14, 15, 16, 18, 23, 26, 28, 31, 34, 35, 40, 42, 43, 63, 64, 71, 75, 79, 82, 83, 86, 87, 88, 98, 102, 107, 108, 112, 115, 126, 127, 129, 135, 136, 138, 141, 142, 146, 150, 151, 154, 155, 160, 166, 170, 174, 181, 184, 190, 191
8	2, 7, 8, 11, 14, 19, 21, 26, 29, 32, 38, 39, 50, 53, 56, 61, 63, 74, 79, 80, 87, 91, 104, 106, 122, 125, 128, 130, 133, 141, 147, 151, 152, 154, 155, 158, 165, 167, 173, 175, 176, 186, 187, 194, 199, 200
9	2, 7, 23, 24, 29, 30, 31, 34, 39, 42, 46, 47, 48, 50, 61, 62, 72, 74, 90, 94, 96, 106, 114, 117, 120, 125, 127, 130, 134, 135, 138, 144, 147, 162, 165, 168, 178, 191, 192
10	2, 10, 13, 16, 21, 27, 34, 40, 45, 50, 59, 64, 66, 79, 82, 83, 88, 90, 91, 92, 98, 101, 103, 111, 112, 118, 130, 131, 134, 136, 138, 139, 143, 147, 150, 154, 160, 170, 174, 175, 182, 184, 189, 190, 194, 197

Figure 9.e: Random formulas valid in S4.

```
> proc : s4_random()
  local i, j, j2, l, a, total;
  begin
    total := 0;
    for i := 2 to 10 do
    begin
      l :== [];
      for j := 1 to 200 do
      begin
        j2 :== (i mult 200) + j;
        a :== rnd_fml(i, [p0,p1,p2,p3], j2);
        if s4::provable(a) then append(l, j);
      end;
      print(i, " : ", l);
      total := total + nops(l);
    end;
    print("total: ", total, " valid and ", 9 mult 200 - total, " nonvalid formulas");
  end; # s4_random
> s4_random();
```

9.5.4 Valid formulas

For all $d \in \{3, 5, 7\}$ we compute 1000 random formulas A with depth d, variables in $\{p_0, \ldots, p_6\}$ and initial value 117 for j, and check whether the formula add_garbage(nnf($\neg A$)) $\vee$ A is valid in S4. Two of the generated formulas are printed in order to show typical generated formulas.

Attention: In the following LWB session we use only 10 random formulas instead of 1000, since otherwise the typesetting process would take too much time. Formula 654 with depth 7 proved to be too hard for the LWB 1.0.

```
> j :== 117;
  for i := 1 to 1000
  do begin
    a :== rnd_fml(3, [p0,p1,p2,p3,p4,p5,p6], j);
    b :== add_garbage(nnf(~a));
    if ( not s4::provable(b v a) ) then print("***** error *****");
  end;
  print("done");
> j :== 117;
  for i := 1 to 1000
  do begin
    a :== rnd_fml(5, [p0,p1,p2,p3,p4,p5,p6], j);
    b :== add_garbage(nnf(~a));
    if ( not s4::provable(b v a) ) then print("***** error *****");
  end;
  print("done");
> j :== 117;
  for i := 1 to 10 # Attention: Use 1000 instead of 10 for your tests.
  do begin
    a :== rnd_fml(7, [p0,p1,p2,p3,p4,p5,p6], j);
    b :== add_garbage(nnf(~a));
    if ( not s4::provable(b v a) ) then print("***** error *****");
  end;
  print("done");
```

9.5.5 Nonvalid formulas

We choose the nonderivability results $\mathsf{KT4B}^{\mathcal{H}} \nvdash 5$, $\mathsf{KT4Dum}^{\mathcal{H}} \nvdash$ Grz, and $\mathsf{KT45}^{\mathcal{H}} \nvdash$ Dum. The 1000 random formulas with depth 3, variables in $\{p_0, p_1, p_2, p_3, p_4\}$, and initial value 234.

```
> for i := 1 to 1000
  do begin
    tuple :== [];
    j :== 234;
    a1 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a2 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a3 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a4 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    b :== A4{a1/p0} & A4{a2/p0} & A4{a3/p0} & A4{a4/p0} -> A5;
    if ( i = 123 ) then print(b);
    if ( s4::provable(b) ) then print("***** error *****");
  end;
> for i := 1 to 1000
  do begin
    tuple :== [];
    j :== 255;
    a1 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a2 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a3 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a4 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    b :== Dum{a1/p0} & Dum{a2/p0} & Dum{a3/p0} & Dum{a4/p0} -> Grz;
    if ( i = 123 ) then print(b);
    if ( s4::provable(b) ) then print("***** error *****");
  end;
> for i := 1 to 1000
  do begin
    tuple :== [];
    j :== 234;
    a1 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a2 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a3 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    a4 :== rnd_fml(3, [p0,p1,p2,p3,p4], j);
    b :== A5{a1/p0} & A5{a2/p0} & A5{a3/p0} & A5{a4/p0} -> Dum;
    if ( i = 123 ) then print(b);
    if ( s4::provable(b) ) then print("***** error *****");
  end;
```

9.6 OTHER LOGICS

Of course every collection of formulas can be used for tests, i.e. also sets of benchmark formulas. However, we are not aware of large collections of formulas dedicated to tests. A good test for IPC decision procedures is the computation of the numbers in the table in theorem 12.6.4.

9.7 SUMMARY

We have described tests for decision procedures for the logics K, KT, S4. For each of these logics, they include several thousand formulas of various structures. Most of these formulas are generated by procedures, which saves time and prevents typing errors.

10

BENCHMARKS

One second is the time occupied by 9’192’631’770 vibrations of the light (of a specified wavelength) emitted by a cesium-133 atom.

Halliday/Resnik. Fundamentals of Physics.

10.1 INTRODUCTION

In the introduction we listed some methods for proof search in the propositional modal logics K, KT, and $\mathsf{S4}$. It is difficult to compare the usefulness of these methods in practice, as in most publications, no or only a few execution times are listed. There are some exceptions like [Cat91] (31 formulas for K, KT, $\mathsf{S4}$) and [Dem95] (38 formulas for $\mathsf{S4}$). However, these formulas are already too simple today to serve as benchmarks. For example it takes about 0.002 seconds to solve the most difficult problem of [Cat91] for K, KT, $\mathsf{S4}$ with the LWB. The formulas in [Dem95] are considerably more difficult, but also there it takes only about 0.035 seconds to solve the most difficult formula with the LWB (hardware as described in section 10.8).

This situation is very unsatisfactory. Therefore, we decided to collect a set of benchmark problems for the propositional modal logics K, KT, and $\mathsf{S4}$.

In classical predicate logic, the often-cited collection of Pelletier [Pel86] has been replaced by the TPTP library [SSY94]. Although this library is quite large, it is still common to choose a dozen of these formulas out of this library and publish the execution times for these formulas. Therefore we decided to give not just a list of formulas, but to present a method that makes it possible to compare different provers. (The producers of TPTP plan a so-called benchmark suite for the next version that should allow the computation of a performance index for automated theorem provers for classical predicate logic; cp. [SSY94].)

The selection of the hard benchmark formulas was guided by the following postulates:

1. Valid as well as nonvalid formulas.
2. Formulas of various structures.
3. Some of the benchmark formulas are hard enough for forthcoming provers.

4. For each formula, the result is already known today.
5. Simple 'tricks' do not help to solve the problems.
6. Execution of the benchmark test does not take too much time.
7. The results can be summarised.

These postulates lead us to the exclusive use of scalable formulas. We test for which parameters n a certain prover can decide whether the formula $A(n)$ is valid or not in less than 100 seconds. The drawback of scalable formulas is their regular structure. If 'by chance' a prover 'recognises' this structure, it can perform extremely well for a certain class of formulas, but still work very badly for slightly different formulas. We try to overcome this problem by hiding their structure with superfluous subformulas, and by presenting a sufficient number of different scalable formulas.

Collecting and constructing scalable formulas – always keeping the seven postulates in mind – is a very time-consuming task. Therefore the number of scalable formulas we propose is not as large as we would like it to be.

The number of formulas is no longer a problem in the recent approach chosen in [GS96] (see also [HS97]). There, formulas in some sort of nested conjunctive normal form were constructed using a random generator. It is possible to choose length, number of clauses, and modal depth such that the resulting formulas are on the edge between being satisfiable and not satisfiable. On the other hand, this method has two drawbacks: the formulas all have a similar structure (cp. postulate 2), and – more serious – the correct result is not known beforehand (cp. postulate 4). In our opinion, a method relying on such formulas cannot replace our method, but could serve as a complementary method.

10.2 POSTULATES

In this section we discuss the seven postulates mentioned in the introduction in more detail. Note that they are widely applicable, not just for the logics for which we present benchmark formulas.

Valid as well as nonvalid formulas.

Often only valid formulas are considered as benchmark formulas. We think that not valid formulas are as interesting as valid ones.

Moreover, algorithms like the Greedy procedure in classical propositional logic (see [KP92], [SLM92]) can recognise many nonvalid formulas in a very short time, but they cannot show that a formula is valid. A benchmark method must also be applicable for such semi-decision procedures.

Formulas of various structures.

It is clear that four or five formulas are not enough to obtain representative results about the performance of a prover. However, also a large number of formulas does not guarantee the quality of the benchmark set. For example it is tempting to use the embedding of IPC in S4 in order to obtain a list of benchmark formulas for S4 from a list of benchmark formulas for IPC, but all the resulting formulas look similar. An even more dubious method it the use of just one class of formulas, for example the pigeonhole formulas.

Things of course look different if one has a certain application in mind and develops a tuned prover for this application, but here we want to measure the overall performance.

Some of the benchmark formulas are difficult enough for forthcoming provers.

It is not enough if the benchmark test is difficult enough for today's provers. The test should still be applicable for much faster computers and improved search methods in the future. Thus the test must contain formulas that are far too difficult for existing provers.

For each formula, the result is already known today.

The correct result, i.e. 'valid' or 'nonvalid', must be known already today. Therefore it must be possible to check the validity or nonvalidity of the formulas with logical methods, and not just with theorem provers. Random formulas do not fulfill this postulate.

Simple 'tricks' do not help.

The addition of a simple trick to the prover should not influence the results.

Assume for example that a benchmark set for K contains formulas without $\Box$ and without $\Diamond$. Such formulas are valid in K iff they are valid in classical propositional logic. If a prover checks at the beginning whether modal operators occur in the formula, then he can apply a fast decision procedure for classical propositional logic in order to solve the problem. Therefore our benchmark test contains no such formulas. The pigeonhole formula for example, is disguised with some 'superfluous' $\Box$ and $\Diamond$. If a prover recognises this disguise, or if it sees that a subproblem is purely classical during the proof search, then we no longer consider it a simple trick.

Of course it is impossible to foresee all such tricks, but at least one should consider as many as possible.

Execution of the benchmark test does not take too much time.

Nobody wants to spend a lot of time to measure the performance of his prover. In particular, the required time should not depend on the prover or on the computer. Therefore, some limits must be used, for example 'if the prover cannot solve the problem in n seconds, then stop'.

Results can be summarised.

A list of hundreds of numbers is not a satisfactory result, since it makes a comparison of several provers almost impossible. Therefore it must be possible to summarise the results. On the other hand, the result should still show some properties of the prover. We think that in most cases it is sensible to give a list of numbers and not just one number in order to describe the performance of the prover. (Remember the semi-decision procedures already mentioned above.)

10.3 BENCHMARK METHOD

10.3.1 Formulas

For each of the logics K, KT, $\mathsf{S4}$ there are 9 valid and 9 nonvalid scalable formulas. The formulas in each class are parametrised by a number in $\mathbb{N}$. We call the jth formula in the ith class of valid and nonvalid formulas $F_{\mathrm{p},i,j}$ and $F_{\mathrm{n},i,j}$. Thus $L \models F_{\mathrm{p},i,j}$ and $L \not\models F_{\mathrm{n},i,j}$ for all $i \in \{1, \ldots, 9\}$, $j \in \mathbb{N}$.

Let $t(C)$ be the time in seconds the prover takes to decide whether or not the formula C is valid, i.e. $t : \mathrm{Fml}_L \to \mathbb{N} \cup \infty$. We compute for each $i \in \{1, \ldots, 9\}$ the numbers $n_{\mathrm{p},i} := \max\{j \mid \forall j' \leq j : t(F_{\mathrm{p},i,j'}) < 100 \text{ seconds}\}$ and $n_{\mathrm{n},i} := \max\{j \mid \forall j' \leq j : t(F_{\mathrm{n},i,j'}) < 100 \text{ seconds}\}$. Thus $F_{\mathrm{p},i,n_{\mathrm{p},i}+1}$ is the first formula in the ith class of valid formulas such that the prover cannot decide its validity in less than 100 seconds ($F_{\mathrm{n},i,n_{\mathrm{n},i}+1}$ analogously).

10.3.2 Timing

It is not possible to describe a timing procedure that is sensible for all provers (think for example of non-deterministic and parallel provers). Therefore it is important that everybody who applies the benchmark method describes exactly how the timing took place. (See section 10.8 for an example.)

If possible, then observe the following conditions in order to make the comparison of different provers easier:

- The timing starts after the start-up of the prover.
- No conversions, for example in negation normal form, before the timing starts.
- The timing includes the construction of the data structure that contains the formula. This is especially important if you put additional information into the data structure before starting the 'real' proof search.
- Make your prover accessible in some way, in order to give people the possibility to check your results.

10.3.3 Presentation of the results

We propose the following form to present your results. With this information it should be possible to check the obtained results. (See section 10.8 for an example of a filled in form.)

- Name of the prover. Short description of the methods used by the prover (or a reference).
- Availability of the prover.
- Short list of features your prover offers besides checking the validity of formulas (or a reference). Examples: Is the prover tuned for a certain application? Has the correctness of the prover been verified? Can one use one prover for many logics? Is a proof or a counter-model generated?
- Hardware that was used for the benchmark test.
- Example of the way you timed your prover.
- Results, i.e. the numbers $n_{\mathrm{p},i}$ and $n_{\mathrm{n},i}$ for all $i \in \{1, \dots, 9\}$.
- Discussion of the results (optional).

10.3.4 What about the postulates?

Our method largely satisfies the seven postulates:

1. We have as many classes of valid formulas as classes of nonvalid formulas.
2. The formulas have various properties (number of variables, modal depth, ...) and origins.
3. Each class contains arbitrarily difficult formulas.
4. It is clear which formulas are valid.
5. We tried ...
6. Applying our benchmark method is rather tiresome if one does not automate it, but it can be done within a reasonable time.
7. For each logic the result consists of a list of 18 numbers.

The main problems are the postulate 5, and, especially for semi-decision procedures, the relatively small number of classes.

10.4 PRESENTATION OF FORMULAS

For each scalable formula we list in the following section:

1. Idea: Why is the formula valid or nonvalid?
2. Hiding: How is the formula 'hidden' in order to make the problem more difficult to solve?
3. Characteristics: Modal depth, number of variables, …
4. An inductive definition of the formulas.
5. The first two formulas and the length, modal depth, variables of the first six formulas. This information is generated by LWB procedures.

We have to make sure that there are no inconsistencies between the mathematical definition of a formula class and the corresponding LWB procedure. Therefore we automatically convert the mathematical definition (i.e. its LaTeX source) into LWB procedures and compare the output of the hand-written and the generated LWB procedures.

10.5 K

10.5.1 *k_branch_p*

Idea: The branching formula is defined in [HM92] in order to prove that in K there are formulas A such that the number of worlds of each model of A is exponential with respect to length(A). We use these formulas plus a negation symbol in front and the additional subformula $\neg\Box^n p_{n \text{ div } 3+1}$ in order to make the formula valid. We assume $n < 100$.
Characteristics: $2n+3$ variables, modal depth $n+1$.

$k\_branch\_p(n) :\equiv \neg(p_{100} \wedge \neg p_{101} \wedge \bigwedge_{i=0,\ldots,n}(\Box^i(bdepth(n) \wedge det(n) \wedge branching(n)))) \vee \neg\Box^n p_{n \text{ div } 3+1}$

$bdepth(n) :\equiv \bigwedge_{i=1,\ldots,n+1}(p_{100+i} \rightarrow p_{99+i})$

$det(n) :\equiv \bigwedge_{i=0,\ldots,n}(p_{100+i} \rightarrow (p_i \rightarrow \Box(p_{100+i} \rightarrow p_i)) \wedge (\neg p_i \rightarrow \Box(p_{100+i} \rightarrow \neg p_i)))$

$branching(n) :\equiv$

$\bigwedge_{i=0,\ldots,n-1}(p_{100+i} \wedge \neg p_{101+i} \rightarrow \Diamond(p_{101+i} \wedge \neg p_{102+i} \wedge p_{i+1}) \wedge \Diamond(p_{101+i} \wedge \neg p_{102+i} \wedge \neg p_{i+1}))$

```
> for i := 1 to 6 do
    print(i, ": ", length(k_branch_p(i)), ", ", modaldepth(k_branch_p(i)), ", ",
          vars(k_branch_p(i)));
> k_branch_p(1);
> k_branch_p(2);
```

10.5.2 *k_branch_n*

Idea: The branching formula as defined in [HM92] (cp. 10.5.1). We assume $n < 100$.
Characteristics: $2n+3$ variables, modal depth $n+1$.

$k\_branch\_n(n) :\equiv \neg(p_{100} \wedge \neg p_{101} \wedge \bigwedge_{i=0,\ldots,n}(\Box^i(bdepth(n) \wedge det(n) \wedge branching(n))))$

bdepth, *det*, *branching* as in 10.5.1.

```
> for i := 1 to 6 do
    print(i, ": ", length(k_branch_n(i)), ", ", modaldepth(k_branch_n(i)), ", ",
          vars(k_branch_n(i)));
> k_branch_n(1);
> k_branch_n(2);
```

10.5.3 *k_d4_p*

Idea: $\mathsf{K} \models \mathrm{D} \wedge 4 \wedge \mathrm{B}\{\neg p_0/p_0\} \rightarrow \mathrm{T}$ (cp. theorem 11.3.4).

Hiding: The left hand side occurs with 1 to n $\Box$, the right hand side occurs just with n $\Box$ in front. $\Box^n\mathrm{T}$ is repeated n times. Additionally there are some superfluous instances, and the whole formula is in negation normal form.

Characteristics: One variable, modal depth $n+3$, in negation normal form. There are no 'complicated' formulas inside a $\Box$ or $\Diamond$.

$$\mathit{k\_d4\_p}(n) :\equiv \mathrm{nnf}(\textstyle\bigvee_{i=1,\ldots,n}(\Box^n\mathrm{T} \vee \neg\Box^i\mathrm{D}_2 \vee \neg\Box^i 4 \vee \neg\Box^i 4\{\Diamond p_0/p_0\} \vee \neg\Box^i\mathrm{B} \vee \neg\Box^i\mathrm{B}\{\neg p_0/p_0\}))$$

```
> for i := 1 to 6 do
    print(i, ": ", length(k_d4_p(i)), ", ", modaldepth(k_d4_p(i)), ", ", vars(k_d4_p(i)));
> k_d4_p(1);
> k_d4_p(2);
```

10.5.4 *k_d4_n*

Idea: $\mathsf{S4} \not\models 5$ and $\mathsf{S4} \models \mathrm{D}$, thus $\mathsf{KDT4}^{\mathcal{H}} \nvdash 5$ (cp. chapter 11).

Hiding: As in 10.5.3.

Characteristics: One variable, modal depth $n+3$, in negation normal form.

$$\begin{aligned}\mathit{k\_d4\_n}(n) :\equiv \mathrm{nnf}(\textstyle\bigvee_{i=1,\ldots,n}(&\Box^n(\Box p_0 \vee \Box\Diamond\neg p_0) \vee \neg\Box^i\mathrm{D}_2 \vee \neg\Box^i 4 \vee \neg\Box^i 4\{\Diamond p_0/p_0\} \vee \neg\Box^i\mathrm{D}\\ &\vee \neg\Box^i 4\{\Diamond p_0 \rightarrow p_0/p_0\} \vee \neg\Box^i 4\{\Box p_0 \rightarrow p_0/p_0\}))\end{aligned}$$

```
> for i := 1 to 6 do
    print(i, ": ", length(k_d4_n(i)), ", ", modaldepth(k_d4_n(i)), ", ", vars(k_d4_n(i)));
> k_d4_n(1);
> k_d4_n(2);
```

10.5.5 *k_dum_p*

Idea: $\mathsf{K} \models 4\{\Box(p_0 \rightarrow \Box p_0) \rightarrow p_0/p_0\} \wedge \Box 4 \wedge \mathrm{Dum} \wedge \mathrm{Dum}\{p_0 \rightarrow \Box p_0/p_0\} \rightarrow \mathrm{Dum}_1$ (cp. theorem 11.3.9).

Hiding: Some of the formulas on the left hand side of the implication occur with 1 to $n-1$ $\Box$ in front, the right hand side occurs just once with n div $2+1$ $\Box$ in front.

Characteristics: One variable, modal depth about $\frac{n}{2}+4$. There are no complicated formulas inside a $\Box$ or a $\Diamond$. Certain subformulas (for example $p_0 \rightarrow \Box p_0$) occur many times in these formulas.

$$\begin{aligned}\mathit{k\_dum\_p}(n) :\equiv &\textstyle\bigwedge_{i=1,\ldots,n \text{ div } 2}(\Box^i(\Box 4 \wedge \mathrm{Dum})) \wedge \neg\Box^{n \text{ div } 2+1}\mathrm{Dum}_1\\ &\rightarrow \Diamond^{n \text{ div } 2+1}\neg(4\{\Box(p_0 \rightarrow \Box p_0) \rightarrow p_0/p_0\} \wedge \Box 4 \wedge \mathrm{Dum} \wedge \mathrm{Dum}\{p_0 \rightarrow \Box p_0/p_0\})\\ &\quad \vee \textstyle\bigvee_{i=n \text{ div } 2+2,\ldots,n-1}(\Diamond^i\neg(\Box 4 \wedge \mathrm{Dum}))\end{aligned}$$

```
> for i := 1 to 6 do
    print(i, ": ", length(k_dum_p(i)), ", ", modaldepth(k_dum_p(i)), ", ", vars(k_dum_p(i)));
> k_dum_p(1);
> k_dum_p(2);
```

10.5.6 *k_dum_n*

Idea: $\mathsf{K4Dum}_4^{\mathcal{H}} \nvdash \mathrm{Dum}$ (cp. chapter 11).

Hiding: As in 10.5.5.

Characteristics: One variable, modal depth $n+5$. Certain subformulas (for example $p_0 \to \Box p_0$) occur many times in these formulas.

$$\begin{aligned} \textit{k\_dum\_n}(n) :\equiv\ & \textstyle\bigwedge_{i=1,\ldots,n \text{ div } 2}(\Box^i(\Box 4 \wedge \mathrm{Dum}_4)) \wedge \neg\Box^{n+1}\mathrm{Dum} \\ & \to \Diamond^{n+1}\neg(4\{\Box(p_0 \to \Box p_0) \to p_0/p_0\} \wedge \Box 4 \wedge \mathrm{Dum}_4 \wedge \mathrm{Dum}_4\{p_0 \to \Box p_0/p_0\}) \\ & \quad \vee \textstyle\bigvee_{i=n \text{ div } 2+2,\ldots,n-1}(\Diamond^i\neg(\Box 4 \wedge \mathrm{Dum}_4)) \end{aligned}$$

```
> for i := 1 to 6 do
    print(i, ": ", length(k_dum_n(i)), ", ", modaldepth(k_dum_n(i)), ", ", vars(k_dum_n(i)));
> k_dum_n(1);
> k_dum_n(2);
```

10.5.7 *k_grz_p*

Idea: $\mathsf{K} \models \Box\mathrm{Grz} \wedge \mathrm{Grz}\{C \wedge 4\{C/p_0\}/p_0\} \to \mathrm{Grz}_1$, where $C \equiv \Box(p_2 \to \Box p_2) \to p_2$ (cp. theorem 11.3.9).

Hiding: Many superfluous instances with three variables, and iterated $\Box$ inside the instances.

Characteristics: 3 (if $n < 5$) or 4 variables, modal depth ≥ 7.

$$\begin{aligned} \textit{k\_grz\_p}(n) :\equiv\ & \Box\mathrm{Grz}\{p_2/p_0\} \wedge \textstyle\bigwedge_{i=1,\ldots,n-1}(l(i)) \wedge \mathrm{Grz}\{C() \wedge 4\{C()/p_0\}/p_0\} \\ & \to \mathrm{Grz}_1\{p_1/p_0\} \vee \mathrm{Grz}_1\{p_2/p_0\} \vee \mathrm{Grz}_1\{p_3/p_0\} \end{aligned}$$

$$l(i) :\equiv \begin{cases} \mathrm{Grz}\{\mathit{l2}(i \text{ div } 4)/p_0\} & i \bmod 4 = 0 \\ \mathrm{Grz}\{\Box\mathit{l2}(i \text{ div } 4) \vee p_1/p_0\} & i \bmod 4 = 1 \\ \mathrm{Grz}\{\Box\mathit{l2}(i \text{ div } 4) \vee p_1 \vee p_2/p_0\} & i \bmod 4 = 2 \\ \mathrm{Grz}\{\Box\mathit{l2}(i \text{ div } 4) \vee p_1 \vee p_2 \vee p_3/p_0\} & \text{otherwise} \end{cases}$$

$$\mathit{l2}(i) :\equiv \begin{cases} \mathsf{false} & i = 0 \\ \Box\mathit{l2}(i-1) \vee p_1 \vee p_2 \vee p_3 \vee p_4 & \text{otherwise} \end{cases}$$

$$C() :\equiv \Box(p_2 \to \Box p_2) \to p_2$$

```
> for i := 1 to 6 do
    print(i, ": ", length(k_grz_p(i)), ", ", modaldepth(k_grz_p(i)), ", ", vars(k_grz_p(i)));
> k_grz_p(1);
> k_grz_p(2);
```

10.5.8 *k_grz_n*

Idea: $\mathsf{KGrz}_1^{\mathcal{H}} \not\vdash \mathrm{Grz}$ (cp. chapter 11).

Hiding: As in 10.5.7.

Characteristics: 3 (if $n < 5$) or 4 variables, modal depth ≥ 7.

$$\begin{aligned} \textit{k\_grz\_n}(n) :\equiv\ & \Box\mathrm{Grz}_1\{p_2/p_0\} \wedge \textstyle\bigwedge_{i=1,\ldots,n-1}(l(i)) \wedge \mathrm{Grz}_1\{C() \wedge 4\{C()/p_0\}/p_0\} \\ & \to \mathrm{Grz}\{p_1/p_0\} \vee \mathrm{Grz}\{p_2/p_0\} \vee \mathrm{Grz}\{p_3/p_0\} \end{aligned}$$

$$l(i) :\equiv \begin{cases} \mathrm{Grz}_1\{\mathit{l2}(i \text{ div } 4)/p_0\} & i \bmod 4 = 0 \\ \mathrm{Grz}_1\{\Box\mathit{l2}(i \text{ div } 4) \vee p_1/p_0\} & i \bmod 4 = 1 \\ \mathrm{Grz}_1\{\Box\mathit{l2}(i \text{ div } 4) \vee p_1 \vee p_2/p_0\} & i \bmod 4 = 2 \\ \mathrm{Grz}_1\{\Box\mathit{l2}(i \text{ div } 4) \vee p_1 \vee p_2 \vee p_3/p_0\} & \text{otherwise} \end{cases}$$

l2, C as in 10.5.7.

```
> for i := 1 to 6 do
    print(i, ": ", length(k_grz_n(i)), ", ", modaldepth(k_grz_n(i)), ", ", vars(k_grz_n(i)));
> k_grz_n(1);
> k_grz_n(2);
```

10.5.9 *k_lin_p*

Idea: $\mathsf{K} \models \mathrm{H}\{p_0 \wedge \Box p_0 \rightarrow p_1/p_0, p_1 \wedge \Box p_1 \rightarrow p_0/p_1\} \rightarrow \mathrm{L}$ (cp. theorem 11.3.5). Thus $\mathsf{K} \models \mathit{H2} \rightarrow \mathrm{L}$, where $\mathit{H2} \equiv \mathrm{H}\{p_0 \wedge \Box p_0 \wedge p_0 \rightarrow p_1/p_0, \neg p_0 \rightarrow \neg(\Box p_1 \wedge p_1)/p_1\}$.

Hiding: Superfluous instances of *H2* with other variables.

Characteristics: $n+1$ variables, modal depth 3.

$$\begin{aligned} \mathit{k\_lin\_p}(n) :\equiv\ & \textstyle\bigvee_{i=1,\ldots,n \text{ div } 3}(\neg \mathit{H2}(p_i, p_{i+1})) \\ & \vee \mathrm{L}\{p_n/p_0, p_n/p_1\} \\ & \vee \textstyle\bigvee_{i=n \text{ div } 3+1,\ldots,n}(\neg \mathit{H2}(p_i, p_{i+1})) \\ \mathit{H2}(A,B) :\equiv\ & \mathrm{H}\{A \wedge \Box A \wedge A \rightarrow B/p_0, \neg A \rightarrow \neg(\Box B \wedge B)/p_1\} \end{aligned}$$

```
> for i := 1 to 6 do
    print(i, ": ", length(k_lin_p(i)), ", ", modaldepth(k_lin_p(i)), ", ", vars(k_lin_p(i)));
> k_lin_p(1);
> k_lin_p(2);
```

10.5.10 *k_lin_n*

Idea: $\mathsf{KL}^{\mathcal{H}} \not\models \mathrm{L}^+$ (cp. chapter 11).

Characteristics: $n+1$ variables, modal depth 2 (if $n=1$) or 3.

$$\begin{aligned} \mathit{k\_lin\_n}(n) :\equiv\ & \textstyle\bigvee_{i=1,\ldots,2n-2}(\neg \mathrm{L}\{\Diamond p_i/p_0, p_{i+1}/p_1\} \vee \neg \mathrm{L}\{p_i \rightarrow \Box p_{i+1}/p_0, p_{i+1}/p_1\}) \\ & \vee \mathrm{L}^+\{p_n/p_0, p_n/p_1\} \\ & \vee \textstyle\bigvee_{i=2n,\ldots,4n-4}(\neg \mathrm{L}\{\Diamond p_i/p_0, p_{i+1}/p_1\} \vee \neg \mathrm{L}\{p_i \rightarrow \Box p_{i+1}/p_0, p_{i+1}/p_1\}) \end{aligned}$$

```
> for i := 1 to 6 do
    print(i, ": ", length(k_lin_n(i)), ", ", modaldepth(k_lin_n(i)), ", ", vars(k_lin_n(i)));
> k_lin_n(1);
> k_lin_n(2);
```

10.5.11 *k_path_p*

Idea: Without hiding, the formula would just be $\Box p_{\text{path\_el}(1,n)} \vee \Diamond(\neg p_{\text{path\_el}(1,n)} \wedge \Box p_{\text{path\_el}(2,n)}) \vee \ldots \vee \Diamond(\neg p_{\text{path\_el}(n-1,n)} \wedge \Box p_{\text{path\_el}}(n,n)) \vee \Diamond^n \neg p_{\text{path\_el}(n,n)}$. It is easy to check the validity of these formulas with backward proof search in a sequent calculus.

Hiding: The formula above is the path from the entry to the exit of a labyrinth. The labyrinth consists of a start point, n levels of 6 points, and an end point (see figure 10.5.11). The formula $\Box p_1 \vee \Box p_3 \vee \Box p_5$ connects the start point with the points 1,3,5 on the first level, and $\Diamond^n \neg p_2 \vee \Diamond^n \neg p_4 \vee \Diamond^n \neg p_6$ connects the points 2,4,6 on the nth level with the end point. The path from the start point to the end point is defined by the formula displayed in the 'idea' part. The formula generated by $\mathit{left\_to\_right}(l,k,n)$ defines with which points on level $l+1$ the point k on level l is connected. The formulas generated by *left_to_right* are the backward connections.

Characteristics: If we do backward proof search in a sequent calculus, then the path from the entry to the exit has to be found with backtracking, i.e. the function *path_el* must be reconstructed. There is only one such path. 6 variables, modal depth n.

$$\begin{aligned} \mathit{k\_path\_p}(n) :\equiv\ & (\Box p_1 \vee \Box p_{\mathit{path\_el}(1,n)} \vee \Box p_3 \vee \Box p_5) \\ & \vee \textstyle\bigvee_{i=1,\ldots,n; j=1,\ldots,6}(\mathit{left\_to\_right}(i,j,n) \vee \mathit{right\_to\_left}(i,j,n)) \\ & \vee (\Diamond^n \neg p_2 \vee \Diamond^n \neg p_4 \vee \Diamond^n \neg p_{\mathit{path\_el}(n,n)} \vee \Diamond^n \neg p_6) \end{aligned}$$

$$\mathit{path\_el}(l,n) :\equiv \begin{cases} 2 & l = 1 \\ \mathit{modg}(\mathit{path\_el}(l-1,n)+3, 6) & l > n \text{ div } 2 \\ \mathit{modg}(\mathit{path\_el}(l-1,n)+5, 6) & \text{otherwise} \end{cases}$$

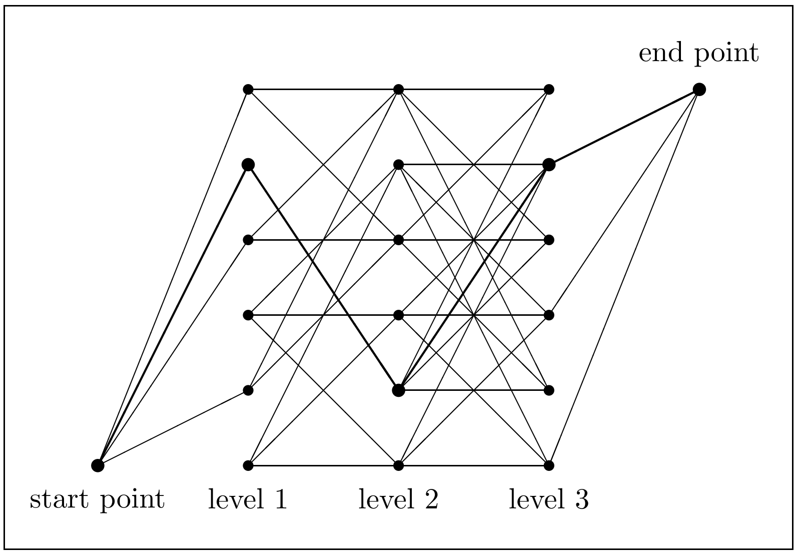


Figure 10.a: The labyrinth given by the formula $k\_path\_p(3)$. The thick line is the path.

$$
left\_to\_right(l,k,n) :\equiv \\
\begin{cases}
\mathsf{false} & l = n \\
\mathsf{false} & l \neq n, k \bmod 2 = 0, k \neq path\_el(l,n) \\
lists2fml(l,k,[path\_el(l+1,n)]) & l \neq n, k \bmod 2 = 0, k = path\_el(l,n) \\
lists2fml(l,k,[1,3,path\_el(l+1,n),5]) & l \neq n, k \bmod 2 \neq 0, k = path\_el(l,n) \\
lists2fml(l,k,delete(path\_el(l+1,n),1,3,5)) & \text{otherwise}
\end{cases}
$$

$$
right\_to\_left(l,k,n) :\equiv \begin{cases}
\mathsf{false} & l = 1 \\
\mathsf{false} & l \neq 1, k \bmod 2 = 1 \\
lists2fml\_back(l-1,k,delete(path\_el(l-1,n),2,4,6)) & \text{otherwise}
\end{cases}
$$

$$lists2fml(l,k,s) :\equiv \bigvee_{x \in s}(\Diamond^l(\neg p_k \wedge \Box p_x))$$

$$lists2fml\_back(l,k,s) :\equiv \bigvee_{x \in s}(\Diamond^l(\neg p_x \wedge \Box p_k))$$

$$
delete(x,y1,y2,y3) :\equiv \begin{cases}
[y2,y3] & x = y1 \\
[y1,y3] & x = y2 \\
[y1,y2] & x = y3 \\
[y1,y2,y3] & \text{otherwise}
\end{cases}
$$

$$
modg(n_1,n_2) :\equiv \begin{cases}
n_2 & n_1 \bmod n_2 = 0 \\
n_1 \bmod n_2 & \text{otherwise}
\end{cases}
$$

```
> for i := 1 to 6 do
    print(i, ": ", length(k_path_p(i)), ", ", modaldepth(k_path_p(i)), ", ", vars(k_path_p(i)));
> k_path_p(1);
> k_path_p(2);
```

10.5.12 *k_path_n*

Idea: As in 10.5.11, but one piece of the path is missing (cp. the different definitions of *left_to_right* in 10.5.11 and 10.5.12).

Hiding: As in 10.5.11.

Characteristics: If we do backward proof search in a sequent calculus, then the labyrinth is searched (using backtracking) for a path from the entry to the exit. 6 variables, modal depth $n+1$.

$$\begin{aligned}\mathit{k\_path\_n}(n) :\equiv\ & (\Box p_1 \vee \Box p_{path\_el(1,n+1)} \vee \Box p_3 \vee \Box p_5)\\ & \vee \bigvee_{i=1,\ldots,n+1;j=1,\ldots,6}(\mathit{left\_to\_right}(i,j,n+1) \vee \mathit{right\_to\_left}(i,j,n+1))\\ & \vee (\Diamond^{n+1}\neg p_2 \vee \Diamond^{n+1}\neg p_4 \vee \Diamond^{n+1}\neg p_{path\_el(n+1,n+1)} \vee \Diamond^{n+1}\neg p_6)\end{aligned}$$

path_el, *right_to_left*, *lists2fml*, *lists2fml_back*, *delete*, *modg* as in 10.5.11.

$$\mathit{left\_to\_right}(l,k,n) :\equiv \begin{cases} \mathsf{false} & l = n\\ \mathsf{false} & l \neq n, k \bmod 2 = 0, k \neq \mathit{path\_el}(l,n)\\ \mathsf{false} & l \neq n, k \bmod 2 = 0, k = \mathit{path\_el}(l,n), l = n \text{ div } 2\\ \mathit{lists2fml}(l,k,[\mathit{path\_el}(l+1,n)]) & l \neq n, k \bmod 2 = 0, k = \mathit{path\_el}(l,n), l \neq n \text{ div } 2\\ \mathit{lists2fml}(l,k,[1,3,5]) & l \neq n, k \bmod 2 \neq 0, k = \mathit{path\_el}(l,n), l = n \text{ div } 2\\ \mathit{lists2fml}(l,k,[1,3,\mathit{path\_el}(l+1,n),5]) & l \neq n, k \bmod 2 \neq 0, k = \mathit{path\_el}(l,n), l \neq n \text{ div } 2\\ \mathit{lists2fml}(l,k,\mathit{delete}(\mathit{path\_el}(l+1,n),1,3,5)) & \text{otherwise}\end{cases}$$

```
> for i := 1 to 6 do
    print(i, ": ", length(k_path_n(i)), ", ", modaldepth(k_path_n(i)), ", ", vars(k_path_n(i)));
> k_path_n(1);
> k_path_n(2);
```

10.5.13 *k_ph_p*

Idea: The pigeonhole formulas, which are valid in CPC and thus in K. We assume $n < 100$.

Hiding: Some $\Box$ and $\Diamond$.

Characteristics: Essentially a CPC problem. $n^2 + n$ variables, modal depth 1 (if $n = 1$) or 2.

$\mathit{k\_ph\_p}(n) :\equiv \Diamond \mathit{left}(n) \rightarrow \Diamond \mathit{right}(n)$

$\mathit{left}(n) :\equiv \bigwedge_{i=1,\ldots,n+1}(\bigvee_{j=1,\ldots,n}(l(i,j)))$

$\mathit{right}(n) :\equiv \bigvee_{j=1,\ldots,n;i_1=1,\ldots,n+1;i_2=i_1+1,\ldots,n+1}(l(i_1,j) \wedge l(i_2,j))$

$$l(i,j) :\equiv \begin{cases} \Box p_{100i+j} & i < j\\ p_{100i+j} & \text{otherwise}\end{cases}$$

```
> for i := 1 to 6 do
    print(i, ": ", length(k_ph_p(i)), ", ", modaldepth(k_ph_p(i)), ", ", vars(k_ph_p(i)));
> k_ph_p(1);
> k_ph_p(2);
```

10.5.14 *k_ph_n*

Idea: The pigeonhole formulas, with one missing conjunct on the right hand side to make them nonvalid. We assume $n < 100$.

Hiding: Some $\Box$ and $\Diamond$.

Characteristics: Essentially a CPC problem. $n^2 + n$ variables, modal depth 1 (if $n = 1$) or 2.

$\mathit{k\_ph\_n}(n) :\equiv \Diamond \mathit{left}(n) \rightarrow \Diamond \mathit{right}(n)$

left, *l* as in 10.5.13.

$\mathit{right}(n) :\equiv \bigvee_{j=1,\ldots,n;i_1=1,\ldots,n+1;i_2=i_1+1,\ldots,n+1}(\mathit{l2}(n,i_1,j) \wedge \mathit{l2}(n,i_2,j))$

$$\mathit{l2}(n,i,j) :\equiv \begin{cases} \neg l(i,j) & i = j, i = (2n) \text{ div } 3 + 1\\ l(i,j) & \text{otherwise}\end{cases}$$

```
> for i := 1 to 6 do
    print(i, ": ", length(k_ph_n(i)), ", ", modaldepth(k_ph_n(i)), ", ", vars(k_ph_n(i)));
> k_ph_n(1);
> k_ph_n(2);
```

10.5.15 *k_poly_p*

Idea: The formula $(p_1 \leftrightarrow p_2) \vee (p_2 \leftrightarrow p_3) \vee \ldots \vee (p_{n-1} \leftrightarrow p_n) \vee (p_n \leftrightarrow p_1)$ says: If we have a polygon with n vertices, and all the vertices are either black or white, then two adjacent vertices have the same colour. If n is odd, then this formula is valid in CPC.

Hiding: Many $\Box$, $\Diamond$, and superfluous subformulas.

Characteristics: Essentially a CPC problem. In contrast to the formulas *k_ph_p*, these formulas have a larger modal depth, but less variables. About $4.5n$ variables, modal depth about $3n$.

$$k\_poly\_p(n) :\equiv \begin{cases} poly(3n+1) & n \bmod 2 = 0 \\ poly(3n) & n \bmod 2 = 1 \end{cases}$$

$$poly(n) :\equiv \Box^{n+1} \bigwedge_{i=1,\ldots,n+1}(p_i) \vee f(n,n) \vee \Box^{n+1} \bigwedge_{i=1,\ldots,n+1}(\neg p_{2i})$$

$$f(i,n) :\equiv \begin{cases} \mathsf{false} & i = 0 \\ \Diamond(f(i-1,n) \vee \Diamond^i(p_n \leftrightarrow p_1)) \vee \Box p_{i+2} & i = n \\ \Diamond(f(i-1,n) \vee \Diamond^i(p_i \leftrightarrow p_{i+1})) \vee \Box p_{i+2} & \text{otherwise} \end{cases}$$

```
> for i := 1 to 6 do
    print(i, ": ", length(k_poly_p(i)), ", ", modaldepth(k_poly_p(i)), ", ", vars(k_poly_p(i)));
> k_poly_p(1);
> k_poly_p(2);
```

10.5.16 *k_poly_n*

Idea: As in 10.5.15, but for an even number of vertices.

Hiding: Many $\Box$, $\Diamond$, and superfluous subformulas. The superfluous subformulas do not influence the nonvalidity.

Characteristics: Essentially a CPC problem. In contrast to the formulas *k_ph_n*, these formulas have a larger modal depth, but less variables. About $4.5n$ variables, modal depth about $3n$.

$$k\_poly\_n(n) :\equiv \begin{cases} poly(3n) & n \bmod 2 = 0 \\ poly(3n+1) & n \bmod 2 = 1 \end{cases}$$

poly, f as in 10.5.15.

```
> for i := 1 to 6 do
    print(i, ": ", length(k_poly_n(i)), ", ", modaldepth(k_poly_n(i)), ", ", vars(k_poly_n(i)));
> k_poly_n(1);
> k_poly_n(2);
```

10.5.17 *k_t4p_p*

Idea: $\mathsf{K} \models \mathrm{T}\{\Diamond p_0/p_0\} \wedge \Box\mathrm{T}\{\neg\Box\Diamond p_0/p_0\} \wedge 4\{\Diamond p_0/p_0\} \wedge \mathrm{P} \rightarrow \mathrm{nnf}(\mathrm{M})$ (cp. theorem 11.3.8).

Hiding: Superfluous subformulas ($\Diamond\neg p_3$, $\Diamond p_4$), superfluous instances of 4 and 5, nested $\Box$.

Characteristics: 4 variables, modal depth $n+5$. Partially in negation normal form.

$$k\_t4p\_p(n) :\equiv E(n) \vee \mathrm{nnf}(\neg C(n)) \vee \Diamond p_4$$

$$C(i) :\equiv \begin{cases} ((\Box\Diamond p_0 \rightarrow \Diamond p_1) \wedge \Box(\Box\neg\Box\Diamond p_1 \rightarrow \neg\Box\Diamond p_0) \wedge (\Box\Diamond p_0 \rightarrow \Box\Box\Diamond p_1) & \\ \quad \wedge\Box(\Diamond\Box\Diamond p_0 \wedge p_0 \rightarrow \Box p_1) \wedge \Box\Diamond p_1)\{p_0 \wedge \Diamond\neg p_3/p_1\} & i = 0 \\ \Box 4\{p_1/p_0\} \wedge \Box C(i-1) \wedge \Box 4\{\Diamond p_1/p_0\} & \text{otherwise} \end{cases}$$

$$E(i) :\equiv \begin{cases} \Diamond\Box p_0 & i = 0 \\ \Diamond\neg 4\{\neg p_1/p_0\} \vee \Box E(i-1) \vee \Box 5\{p_1/p_0\} & \text{otherwise} \end{cases}$$

```
> for i := 1 to 6 do
    print(i, ": ", length(k_t4p_p(i)), ", ", modaldepth(k_t4p_p(i)), ", ", vars(k_t4p_p(i)));
```

```
> k_t4p_p(1);
> k_t4p_p(2);
```

10.5.18 *k_t4p_n*

Idea: $\mathsf{KT4P}^{\mathcal{H}} \nvdash \Diamond\Box p_0$ (cp. theorem 11.3.8).

Hiding: As in 10.5.17.

Characteristics: 4 variables, modal depth $2n+4$. Partially in negation normal form.

$$k\_t4p\_n(n) :\equiv E(2n-1) \vee \mathrm{nnf}(\neg C(2n-1)) \vee \Diamond p_4$$

$$C(i) :\equiv \begin{cases} ((\Box\Diamond p_0 \to \Diamond p_1) \wedge \Box(\Box\neg\Box\Diamond p_1 \to \neg\Box\Diamond p_0) \wedge (\Box\Diamond p_0 \to \Box\Box\Diamond p_1) & \\ \quad \wedge\Box(\Diamond\Box\Diamond p_0 \wedge p_0 \to \Box p_1))\{p_0 \wedge \Diamond\neg p_3/p_1\} & i=0 \\ \Box 4\{p_1/p_0\} \wedge \Box C(i-1) \wedge \Box 4\{\Diamond p_1/p_0\} & \text{otherwise} \end{cases}$$

E as in 10.5.17.

```
> for i := 1 to 6 do
    print(i, ": ", length(k_t4p_n(i)), ", ", modaldepth(k_t4p_n(i)), ", ", vars(k_t4p_n(i)));
> k_t4p_n(1);
> k_t4p_n(2);
```

10.6 KT

10.6.1 *kt_45_p*

Idea: $\mathsf{KT}^{\mathcal{H}} \vdash 5\{\Box p_0/p_0\} \wedge \Box 5\{\neg p_0/p_0\} \to 4$ (cp. theorem 11.3.4).

Hiding: The left hand side occurs with 1 to n $\Box$, the right hand side occurs just with n $\Box$ in front. Additionally some superfluous instances, and the whole formula in negation normal form.

Characteristics: One variable, modal depth $n+3$, in negation normal form.

$$kt\_45\_p(n) :\equiv \mathrm{nnf}(\textstyle\bigvee_{i=1,\ldots,n}(\Box^n 4 \vee \neg\Box^i \mathrm{D}_2 \vee \neg\Box^i 5\{\Diamond\neg p_0/p_0\} \vee \neg\Box^i\Box 5 \vee \neg\Box^i \mathrm{B}))$$

```
> for i := 1 to 6 do
    print(i, ": ", length(kt_45_p(i)), ", ", modaldepth(kt_45_p(i)), ", ", vars(kt_45_p(i)));
> kt_45_p(1);
> kt_45_p(2);
```

10.6.2 *kt_45_n*

Idea: $\mathsf{S4} \not\models 5$.

Hiding: As in 10.6.1.

Characteristics: One variable, modal depth $n+3$, in negation normal form.

$$kt\_45\_n(n) :\equiv \mathrm{nnf}(\textstyle\bigvee_{i=1,\ldots,n}(\Box^n(\Box p_0 \vee \Box\Diamond\neg p_0) \vee \neg\Box^i 4 \vee \neg\Box^i 4\{\Diamond p_0/p_0\} \vee \neg\Box^i \mathrm{T} \\ \vee \neg\Box^i 4\{\Diamond p_0 \to p_0/p_0\} \vee \neg\Box^i 4\{\Box p_0 \to p_0/p_0\}))$$

```
> for i := 1 to 6 do
    print(i, ": ", length(kt_45_n(i)), ", ", modaldepth(kt_45_n(i)), ", ", vars(kt_45_n(i)));
> kt_45_n(1);
> kt_45_n(2);
```

10.6.3 *kt_branch_p*

Idea: The branching formula as described in [HM92], plus a negation symbol in front and the additional subformula $\neg\Box^n p_{(n \text{ div } 3)+1}$ in order to make the formula valid.

Characteristics: $2n+3$ variables, modal depth $n+1$. The formulas are much shorter than the corresponding one for K, since in KT the formula $\Box^n A$ is equivalent to $A \wedge \Box A \wedge \ldots \wedge \Box^n A$.

$$\mathit{kt\_branch\_p}(n) :\equiv \neg(p_{100} \wedge \neg p_{101} \wedge \Box^n(\mathit{bdepth}(n) \wedge \mathit{det}(n) \wedge \mathit{branching}(n))) \vee \neg\Box^n p_{(n \text{ div } 3)+1}$$

bdepth, *det*, *branching* as in 10.5.1.

```
> for i := 1 to 6 do
    print(i, ": ", length(kt_branch_p(i)), ", ", modaldepth(kt_branch_p(i)), ", ",
          vars(kt_branch_p(i)));
> kt_branch_p(1);
> kt_branch_p(2);
```

10.6.4 *kt_branch_n*

Idea: The branching formula as defined in [HM92].

Characteristics: Every KT model that satisfies the pure branching formula must have an exponential number of worlds (cp. [HM92]). $2n+3$ variables, modal depth $n+1$. The formulas are much shorter than the corresponding one for K, since in KT the formula $\Box^n A$ is equivalent to $A \wedge \Box A \wedge \ldots \wedge \Box^n A$.

$$\mathit{kt\_branch\_n}(n) :\equiv \neg(p_{100} \wedge \neg p_{101} \wedge \Box^n(\mathit{bdepth}(n) \wedge \mathit{det}(n) \wedge \mathit{branching}(n)))$$

bdepth, *det*, *branching* as in 10.5.1.

```
> for i := 1 to 6 do
    print(i, ": ", length(kt_branch_n(i)), ", ", modaldepth(kt_branch_n(i)), ", ",
          vars(kt_branch_n(i)));
> kt_branch_n(1);
> kt_branch_n(2);
```

10.6.5 *kt_dum_p*

Idea: KT $\models 4\{\Box(p_0 \to \Box p_0) \to p_0\} \wedge \Box 4 \wedge \mathrm{Dum} \wedge \mathrm{Dum}\{p_0 \to \Box p_0/p_0\} \to \mathrm{Dum}_1$ (cp. theorem 11.3.4).

Hiding: Some of he formulas on the left hand side of the implication occur with 1 to $n-1$ $\Box$ in front, the right hand side occurs just once with $(n \text{ div } 2)+1$ $\Box$ in front.

Characteristics: One variable, modal depth $n+1$ (if $n>6$). Certain subformulas (for example $p_0 \to \Box p_0$) occur many times in these formulas.

$$\begin{aligned}\mathit{kt\_dum\_p}(n) :\equiv{} & \textstyle\bigwedge_{i=1,\ldots,n \text{ div } 2}(\Box^i 4) \wedge \neg\Box^{(n \text{ div } 2)+1}\mathrm{Dum}_1\\ & \to \Diamond^{(n \text{ div } 2)+1}\neg(4\{\Box(p_0 \to \Box p_0) \to p_0/p_0\} \wedge \Box 4 \wedge \mathrm{Dum} \wedge \mathrm{Dum}\{p_0 \to \Box p_0/p_0\})\\ & \quad \vee \textstyle\bigvee_{i=n \text{ div } 2+2,\ldots,n-1}(\Diamond^i \neg 4)\end{aligned}$$

```
> for i := 1 to 6 do
    print(i, ": ", length(kt_dum_p(i)), ", ", modaldepth(kt_dum_p(i)), ", ",
          vars(kt_dum_p(i)));
> kt_dum_p(1);
> kt_dum_p(2);
```

10.6.6 *kt_dum_n*

Idea: $\mathsf{KTDum}^{\mathcal{H}} \not\vdash \mathrm{Grz}$ (cp. chapter 11).

Hiding: As in 10.6.5.

Characteristics: One variable, modal depth about $\frac{n}{2}+4$. Certain subformulas (for example $p_0 \to \Box p_0$) occur many times in these formulas.

$$\begin{aligned} \mathit{kt\_dum\_n}(n) :\equiv {} & \textstyle\bigwedge_{i=1,\dots,n \text{ div } 2}(\Box^i(\Box 4 \wedge \mathrm{Dum})) \wedge \neg\Box^{n \text{ div } 2+1}\mathrm{Grz} \\ & \to \Diamond^{n \text{ div } 2+1}\neg(4\{\Box(p_0 \to \Box p_0) \to p_0/p_0\} \wedge \Box 4 \wedge \mathrm{Dum} \wedge \mathrm{Dum}\{p_0 \to \Box p_0/p_0\}) \\ & \quad \vee \textstyle\bigvee_{i=n \text{ div } 2+2,\dots,n-1}(\Diamond^i\neg(\Box 4 \wedge \mathrm{Dum})) \end{aligned}$$

```
> for i := 1 to 6 do
    print(i, ": ", length(kt_dum_n(i)), ", ", modaldepth(kt_dum_n(i)), ", ",
          vars(kt_dum_n(i)));
> kt_dum_n(1);
> kt_dum_n(2);
```

10.6.7 *kt_grz_p*

Idea: $\mathsf{KT} \models \Box\mathrm{Grz} \wedge \mathrm{Grz}\{C \wedge (4\{C/p_0\})/p_0\} \to \mathrm{Grz}_1$, where $C \equiv \Box(p_2 \to \Box p_2) \to p_2$ (cp. theorem 11.3.9).

Hiding: Many superfluous instances with iterated $\Box$ inside the instances. The subformulas $\Diamond\Box\neg p_0$ and $\Diamond p_0$ are the parts of an instance of T.

Characteristics: 3 (if $n < 5$) or 4 variables, modal depth ≥ 7.

$$\begin{aligned} \mathit{kt\_grz\_p}(n) :\equiv {} & \Box\mathrm{Grz}\{p_2/p_0\} \wedge \textstyle\bigwedge_{i=1,\dots,n-1}(l(i)) \wedge \mathrm{Grz}\{C() \wedge (4\{C()/p_0\})/p_0\} \\ & \to \mathrm{Grz}_1\{p_1/p_0\} \vee \mathrm{Grz}_1\{p_2/p_0\} \wedge \Diamond\Box\neg p_0 \vee \mathrm{Grz}_1\{p_3/p_0\} \vee \mathrm{Grz}_1\{p_2/p_0\} \wedge \Diamond p_0 \end{aligned}$$

l, $l2$, C as in 10.5.7.

```
> for i := 1 to 6 do
    print(i, ": ", length(kt_grz_p(i)), ", ", modaldepth(kt_grz_p(i)), ", ",
          vars(kt_grz_p(i)));
> kt_grz_p(1);
> kt_grz_p(2);
```

10.6.8 *kt_grz_n*

Idea: $\mathsf{KTGrz}_1^{\mathcal{H}} \not\vdash 5$ (cp. chapter 11).

Hiding: As in 10.5.7.

Characteristics: 3 (if $n < 5$) or 4 variables, modal depth ≥ 7.

$$\begin{aligned} \mathit{kt\_grz\_n}(n) :\equiv {} & \Box\mathrm{Grz}_1\{p_2/p_0\} \wedge \textstyle\bigwedge_{i=1,\dots,n-1}(l(i)) \wedge \mathrm{Grz}_1\{C() \wedge 4\{C()/p_0\}/p_0\} \\ & \to 5\{p_1/p_0\} \vee 5\{p_2/p_0\} \vee 5\{p_3/p_0\} \end{aligned}$$

$$l(i) :\equiv \begin{cases} \mathrm{Grz}_1\{l2(i \text{ div } 4)/p_0\} & i \bmod 4 = 0 \\ \mathrm{Grz}_1\{\Box l2(i \text{ div } 4) \vee p_1/p_0\} & i \bmod 4 = 1 \\ \mathrm{Grz}_1\{\Box l2(i \text{ div } 4) \vee p_1 \vee p_2/p_0\} & i \bmod 4 = 2 \\ \mathrm{Grz}_1\{\Box l2(i \text{ div } 4) \vee p_1 \vee p_2 \vee p_3/p_0\} & \text{otherwise} \end{cases}$$

$l2$, C as in 10.5.7.

```
> for i := 1 to 6 do
    print(i, ": ", length(kt_grz_n(i)), ", ", modaldepth(kt_grz_n(i)), ", ",
          vars(kt_grz_n(i)));
> kt_grz_n(1);
> kt_grz_n(2);
```

10.6.9 *kt_md_p*

Idea: When doing backward proof search in a sequent calculus, we have to find the way to $g(n, n, \neg p_1)$ through several $\Box$ and $\Diamond$.

Characteristics: One (if $n = 1$) or two variables, modal depth 0 (if $n = 1$) or $n^2 - n + 1$.

$$\begin{aligned}&\mathit{kt\_md\_p}(n) :\equiv\\&\quad p_1 \lor \bigvee_{i=1,\dots,n-1}(g(i,n,\neg p_1 \land \Diamond f(1,n,p_2)) \lor g(i,n,\neg p_1 \land \Diamond f(1,n,p_1)) \lor g(i,n,\neg p_2 \land \Diamond f(1,n,p_1)))\\&\quad \lor g(n,n,\neg p_1)\end{aligned}$$

$$g(i,n,A) :\equiv \begin{cases} A & i = 1 \\ f(i,n,g(i-1,n,A)) & \text{otherwise}\end{cases}$$

$$f(i,n,A) :\equiv \Diamond^{i-1}\Box\Diamond^{n-i}A$$

```
> for i := 1 to 6 do
    print(i, ": ", length(kt_md_p(i)), ", ", modaldepth(kt_md_p(i)), ", ", vars(kt_md_p(i)));
> kt_md_p(1);
> kt_md_p(2);
```

10.6.10 *kt_md_n*

Idea: Similar to 10.6.9, but simpler. The subformula $g(n, n, \neg p_1)$ is missing in order to make the formulas nonvalid.

Characteristics: One variable, modal depth 0 (if $n = 1$) or $n^2 - n + 1$.

$$\mathit{kt\_md\_n}(n) :\equiv p_1 \lor \bigvee_{i=1,\dots,n-1}(g(i,n,\neg p_1 \land \Diamond f(1,n,p_1)))$$

f, g as in 10.6.9.

```
> for i := 1 to 6 do
    print(i, ": ", length(kt_md_n(i)), ", ", modaldepth(kt_md_n(i)), ", ", vars(kt_md_n(i)));
> kt_md_n(1);
> kt_md_n(2);
```

10.6.11 *kt_path_p*

Idea, hiding: As in 10.5.11, but we use new variables on each level of the labyrinth. (The formulas *k_path_p* collapse in KT since KT $\models \Box^n A \to A$.)

Characteristics: $6n$ variables, modal depth n.

$$\begin{aligned}\mathit{kt\_path\_p}(n) :\equiv\ &(\Box p_{11} \lor \Box p_{10+\mathit{path\_el}(1,n)} \lor \Box p_{13} \lor \Box p_{15})\\&\lor \bigvee_{i=1,\dots,n;j=1,\dots,6}(\mathit{left\_to\_right}(i,j,n) \lor \mathit{right\_to\_left}(i,j,n))\\&\lor (\Diamond^n\neg p_{10n+2} \lor \Diamond^n\neg p_{10n+4} \lor \Diamond^n\neg p_{10n+\mathit{path\_el}(n,n)} \lor \Diamond^n\neg p_{10n+6})\end{aligned}$$

path_el, *left_to_right*, *right_to_left*, *delete*, *modg* as in 10.5.11.

$$\mathit{lists2fml}(l,k,s) :\equiv \bigvee_{x\in s}(\Diamond^l(\neg p_{10l+k} \land \Box p_{10(l+1)+x}))$$

$$\mathit{lists2fml\_back}(l,k,s) :\equiv \bigvee_{x\in s}(\Diamond^l(\neg p_{10l+x} \land \Box p_{10(l+1)+k}))$$

```
> for i := 1 to 6 do
    print(i, ": ", length(kt_path_p(i)), ", ", modaldepth(kt_path_p(i)), ", ",
          vars(kt_path_p(i)));
> kt_path_p(1);
> kt_path_p(2);
```

10.6.12 *kt_path_n*

Idea, hiding: As in 10.5.12, but we use new variables on each level of the labyrinth. (The formulas *k_path_p* collapse in KT since KT $\models \Box^n A \to A$.)

Characteristics: $6n$ variables, modal depth $n+1$.

$$\begin{aligned}&\mathit{kt\_path\_n}(n) :\equiv\\ &\quad(\Box p_{11} \vee \Box p_{10+\mathit{path\_el}(1,n+1)} \vee \Box p_{13} \vee \Box p_{15})\\ &\quad \vee \bigvee_{i=1,\ldots,n+1;j=1,\ldots,6}(\mathit{left\_to\_right}(i,j,n+1) \vee \mathit{right\_to\_left}(i,j,n+1))\\ &\quad \vee (\Diamond^{n+1}\neg p_{10(n+1)+2} \vee \Diamond^{n+1}\neg p_{10(n+1)+4} \vee \Diamond^{n+1}\neg p_{10(n+1)+\mathit{path\_el}(n+1,n+1)} \vee \Diamond^{n+1}\neg p_{10(n+1)+6})\end{aligned}$$

path_el, *left_to_right*, *right_to_left*, *delete*, *modg* as in 10.5.12.

lists2fml, *lists2fml_back* as in 10.6.11.

```
> for i := 1 to 6 do
    print(i, ": ", length(kt_path_n(i)), ", ", modaldepth(kt_path_n(i)), ", ",
          vars(kt_path_n(i)));
> kt_path_n(1);
> kt_path_n(2);
```

10.6.13 *kt_ph_p*

Idea, hiding: As in 10.5.13.

Characteristics: Essentially a CPC problem. n^2+n variables, modal depth 1 (if $n=1$) or 2.

$\mathit{kt\_ph\_p}(n) :\equiv \mathit{left}(n) \to \Diamond \mathit{right}(n)$

left, *right*, *l* as in 10.5.13.

```
> for i := 1 to 6 do
    print(i, ": ", length(kt_ph_p(i)), ", ", modaldepth(kt_ph_p(i)), ", ", vars(kt_ph_p(i)));
> kt_ph_p(1);
> kt_ph_p(2);
```

10.6.14 *kt_ph_n*

Idea, hiding: As in 10.5.14.

Characteristics: Essentially a CPC problem. n^2+n variables, modal depth 1 (if $n=1$) or 2.

$\mathit{kt\_ph\_n}(n) :\equiv \mathit{left}(n) \to \Diamond \mathit{right}(n)$

left, *right*, *l*, *l2* as in 10.5.14.

```
> for i := 1 to 6 do
    print(i, ": ", length(kt_ph_n(i)), ", ", modaldepth(kt_ph_n(i)), ", ", vars(kt_ph_n(i)));
> kt_ph_n(1);
> kt_ph_n(2);
```

10.6.15 *kt_poly_p*

Idea, hiding: As in 10.5.15.

Characteristics: Essentially a CPC problem. About $7.5n$ variables, modal depth about $5n$.

$$\mathit{kt\_poly\_p}(n) :\equiv \begin{cases} \mathit{poly}(5n+1) & n \bmod 2 = 0\\ \mathit{poly}(5n) & n \bmod 2 = 1\end{cases}$$

poly as in 10.5.15.

$$f(i,n) :\equiv \begin{cases} \mathsf{false} & i=0\\ \Diamond(f(i-1,n) \vee \Diamond^{i+2}(p_n \leftrightarrow p_1)) \vee \Box p_{i+2} & i=n\\ \Diamond(f(i-1,n) \vee \Diamond^{i+2}(p_i \leftrightarrow p_{i+1})) \vee \Box p_{i+2} & \text{otherwise}\end{cases}$$

```
> for i := 1 to 6 do
    print(i, ": ", length(kt_poly_p(i)), ", ", modaldepth(kt_poly_p(i)), ", ",
          vars(kt_poly_p(i)));
> kt_poly_p(1);
> kt_poly_p(2);
```

10.6.16 *kt_poly_n*

Idea, hiding: As in 10.5.16.

Characteristics: Essentially a CPC problem. About $7.5n$ variables, modal depth about $5n$.

$$kt\_poly\_n(n) :\equiv \begin{cases} poly(3n) & n \bmod 2 = 0 \\ poly(3n+1) & n \bmod 2 = 1 \end{cases}$$

poly, f as in 10.6.15.

```
> for i := 1 to 6 do
    print(i, ": ", length(kt_poly_n(i)), ", ", modaldepth(kt_poly_n(i)), ", ",
          vars(kt_poly_n(i)));
> kt_poly_n(1);
> kt_poly_n(2);
```

10.6.17 *kt_t4p_p*

Idea, hiding: As in 10.5.17.

Characteristics: 4 variables, modal depth $n+4$. Partially in negation normal form.

$kt\_t4p\_p(n) :\equiv E(n) \lor \mathrm{nnf}(\neg C(n)) \lor \Diamond p_4$

$$C(i) :\equiv \begin{cases} ((\Box\Diamond p_0 \to \Box\Box\Diamond p_1) \land \Box(\Diamond\Box\Diamond p_0 \land p_0 \to \Box p_1) \land \Box\Diamond p_1)\{p_0 \land \Diamond\neg p_3/p_1\} & i = 0 \\ \Box 4\{p_1/p_0\} \land \Box C(i-1) & \text{otherwise} \end{cases}$$

E as in 10.5.17.

```
> for i := 1 to 6 do
    print(i, ": ", length(kt_t4p_p(i)), ", ", modaldepth(kt_t4p_p(i)), ", ",
          vars(kt_t4p_p(i)));
> kt_t4p_p(1);
> kt_t4p_p(2);
```

10.6.18 *kt_t4p_n*

Idea, hiding: As in 10.5.18.

Characteristics: 4 variables, modal depth $2n+3$. Partially in negation normal form.

$kt\_t4p\_n(n) :\equiv E(2n-1) \lor \mathrm{nnf}(\neg C(2n-1)) \lor \Diamond p_4$

$$C(i) :\equiv \begin{cases} ((\Box\Diamond p_0 \to \Diamond p_1) \land (\Box\Diamond p_0 \to \Box\Box\Diamond p_1) & \\ \quad \land\Box(\Diamond\Box\Diamond p_0 \land p_0 \to \Box p_1))\{p_0 \land \Diamond\neg p_3/p_1\} & i = 0 \\ \Box 4\{p_1/p_0\} \land \Box C(i-1) & \text{otherwise} \end{cases}$$

E as in 10.5.17.

```
> for i := 1 to 6 do
    print(i, ": ", length(kt_t4p_n(i)), ", ", modaldepth(kt_t4p_n(i)), ", ",
          vars(kt_t4p_n(i)));
> kt_t4p_n(1);
> kt_t4p_n(2);
```

10.7 S4

10.7.1 *s4_45_p*

Idea: S4 $\models 5\{\Box p_0/p_0\} \wedge \Box 5\{\neg p_0/p_0\} \rightarrow 5$.

Hiding: The left hand side occurs with 1 to n $\Box$, the right hand side occurs just with n $\Box$ in front. Additionally some superfluous instances, and the whole formula is in negation normal form.

Characteristics: Two variables, modal depth $n+3$, in negation normal form.

$$\mathit{s4\_45\_p}(n) :\equiv \mathrm{nnf}(\textstyle\bigvee_{i=1,\ldots,n}(h(n,5) \vee \neg h(i,\mathrm{D}_2) \vee \neg h(i,5\{\Diamond\neg p_0/p_0\}) \vee \neg h(i,\Box 5) \vee \neg h(i,\mathrm{B})))$$

$$h(i,A) :\equiv \begin{cases} A & i=0 \\ \Box p_0 \vee \Box h(i-1,A) \vee \Box p_1 & \text{otherwise} \end{cases}$$

```
> for i := 1 to 6 do
    print(i, ": ", length(s4_45_p(i)), ", ", modaldepth(s4_45_p(i)), ", ", vars(s4_45_p(i)));
> s4_45_p(1);
> s4_45_p(2);
```

10.7.2 *s4_45_n*

Idea: S4 $\not\models 5$.

Hiding: As in 10.7.1.

Characteristics: Two variables, modal depth $n+3$, in negation normal form.

$$\mathit{s4\_45\_n}(n) :\equiv \mathrm{nnf}(\textstyle\bigvee_{i=1,\ldots,n}(h(n,(\Box p_0 \vee \Box\Diamond\neg p_0)) \vee \neg h(i,4) \vee \neg h(i,4\{\Diamond p_0/p_0\}) \vee \neg h(i,\mathrm{T}) \\ \vee \neg h(i,4\{\Box p_0 \rightarrow p_0/p_0\}) \vee \neg h(i,\mathrm{T}\{\Box p_0 \rightarrow p_0/p_0\})))$$

h as in 10.7.1.

```
> for i := 1 to 6 do
    print(i, ": ", length(s4_45_n(i)), ", ", modaldepth(s4_45_n(i)), ", ", vars(s4_45_n(i)));
> s4_45_n(1);
> s4_45_n(2);
```

10.7.3 *s4_branch_p*

Idea: The branching formula as described in [HM92], plus a negation symbol in front and the additional subformula $\Box p_{(n \text{ div } 3)+1}$ in order to make the formula valid.

Characteristics: $2n+3$ variables, modal depth 2.

$$\mathit{s4\_branch\_p}(n) :\equiv \neg(p_{100} \wedge \neg p_{101} \wedge \Box(\mathit{bdepth}(n) \wedge \mathit{det}(n) \wedge \mathit{branching}(n))) \vee \neg\Box p_{(n \text{ div } 3)+1}$$

bdepth, *det*, *branching* as in 10.5.1.

```
> for i := 1 to 6 do
    print(i, ": ", length(s4_branch_p(i)), ", ", modaldepth(s4_branch_p(i)), ", ",
          vars(s4_branch_p(i)));
> s4_branch_p(1);
> s4_branch_p(2);
```

10.7.4 *s4_branch_n*

Idea: The branching formula as defined in [HM92].

Characteristics: Every S4 model that satisfies the pure branching formula must have an exponential number of worlds (cp. [HM92]). $2n + 3$ variables, modal depth 2.

$$s4\_branch\_n(n) :\equiv \neg(p_{100} \wedge \neg p_{101} \wedge \Box(bdepth(n) \wedge det(n) \wedge branching(n)))$$

bdepth, *det*, *branching* as in 10.5.1.

```
> for i := 1 to 6 do
    print(i, ": ", length(s4_branch_n(i)), ", ", modaldepth(s4_branch_n(i)), ", ",
          vars(s4_branch_n(i)));
> s4_branch_n(1);
> s4_branch_n(2);
```

10.7.5 *s4_grz_p*

Idea: $\mathsf{S4} \vdash \Box\mathrm{Grz} \wedge \mathrm{Grz}\{C \wedge (4\{C/p_0\})/p_0 \rightarrow \mathrm{Grz}_1$, where $C \equiv \Box(p_2 \rightarrow \Box p_2) \rightarrow p_2$ (cp. theorem 11.3.9).

Hiding: Many superfluous instances with iterated $\Box$ inside the instances. The subformulas $\neg\Box\Diamond p_0$ and $\neg\Diamond\neg(\Box\Diamond p_0 \vee p_1)$ are the parts of a weakened instance of *4*.

Characteristics: 4 (if $n < 5$) or 5 variables, modal depth ≥ 7.

$$\begin{aligned} &s4\_grz\_p(n) :\equiv \\ &\quad \Box\mathrm{Grz}\{p_2/p_0\} \wedge \textstyle\bigwedge_{i=1,\ldots,n-1}(l(i)) \wedge \mathrm{Grz}\{C() \wedge (4\{C()/p_0\})/p_0\} \\ &\quad \rightarrow \mathrm{Grz}_1\{p_1/p_0\} \vee \mathrm{Grz}_1\{p_2/p_0\} \wedge \neg\Box\Diamond p_0 \vee \mathrm{Grz}_1\{p_3/p_0\} \vee \mathrm{Grz}_1\{p_2/p_0\} \wedge \neg\Diamond\neg(\Box\Diamond p_0 \vee p_1) \end{aligned}$$

l, *l2*, *C* as in 10.5.7.

```
> for i := 1 to 6 do
    print(i, ": ", length(s4_grz_p(i)), ", ", modaldepth(s4_grz_p(i)), ", ",
          vars(s4_grz_p(i)));
> s4_grz_p(1);
> s4_grz_p(2);
```

10.7.6 *s4_grz_n*

Idea: $\mathsf{S4Grz}_1^{\mathcal{H}} \nvdash 5$ (cp. chapter 11).

Hiding: As in 10.5.7.

Characteristics: 3 (if $n < 5$) or 4 variables, modal depth ≥ 7.

$$\begin{aligned} s4\_grz\_n(n) :\equiv\ &\Box\mathrm{Grz}_1\{p_2/p_0\} \wedge \textstyle\bigwedge_{i=1,\ldots,n-1}(l(i)) \wedge \mathrm{Grz}_1\{C() \wedge 4\{C()/p_0\}/p_0\} \\ &\rightarrow 5\{p_1/p_0\} \vee 5\{p_2/p_0\} \vee 5\{p_3/p_0\} \end{aligned}$$

$$l(i) :\equiv \begin{cases} \mathrm{Grz}_1\{l2(i \text{ div } 4)/p_0\} & i \bmod 4 = 0 \\ \mathrm{Grz}_1\{\Box l2(i \text{ div } 4) \vee p_1/p_0\} & i \bmod 4 = 1 \\ \mathrm{Grz}_1\{\Box l2(i \text{ div } 4) \vee p_1 \vee p_2/p_0\} & i \bmod 4 = 2 \\ \mathrm{Grz}_1\{\Box l2(i \text{ div } 4) \vee p_1 \vee p_2 \vee p_3/p_0\} & \text{otherwise} \end{cases}$$

l2, *C* as in 10.5.7.

```
> for i := 1 to 6 do
    print(i, ": ", length(s4_grz_n(i)), ", ", modaldepth(s4_grz_n(i)), ", ",
          vars(s4_grz_n(i)));
> s4_grz_n(1);
> s4_grz_n(2);
```

10.7.7 *s4_ipc_p*

Idea: We embed the formula $\bigwedge_{i=1,\ldots,n}(p_1 \wedge \ldots \wedge p_n \rightarrow \mathsf{false}) \rightarrow \mathsf{false}$, which is valid in IPC, in the logic S4.

Characteristics: n variables, modal depth 3.

$s4\_ipc\_p(n) :\equiv \bigwedge_{i=1,\ldots,n}(f(i,n)) \rightarrow \mathsf{false}$

$f(i,n) :\equiv \Box(\Box(\Box p_i \rightarrow \bigwedge_{j=1,\ldots,n}(\Box p_j)) \rightarrow \mathsf{false})$

```
> for i := 1 to 6 do
    print(i, ": ", length(s4_ipc_p(i)), ", ", modaldepth(s4_ipc_p(i)), ", ",
          vars(s4_ipc_p(i)));
> s4_ipc_p(1);
> s4_ipc_p(2);
```

10.7.8 *s4_ipc_n*

Idea: We take the formula of IPC from 10.7.7, remove one part in order to make it nonvalid, and embed it in S4.

Characteristics: n variables, modal depth 3.

$s4\_ipc\_n(n) :\equiv \bigwedge_{i=1,\ldots,n}(f2(i,n)) \rightarrow \mathsf{false}$

$$f2(i,n) :\equiv \begin{cases} \mathsf{true} & i = (n+1) \text{ div } 2 \\ f(i,n) & \text{otherwise} \end{cases}$$

f as in 10.7.7.

```
> for i := 1 to 6 do
    print(i, ": ", length(s4_ipc_n(i)), ", ", modaldepth(s4_ipc_n(i)), ", ", vars(s4_ipc_n(i)));
> s4_ipc_n(1);
> s4_ipc_n(2);
```

10.7.9 *s4_md_p*

Idea: When doing backward proof search, we have to find the way to $g(n,n,\neg p_1)$ through a lot of $\Box$ and $\Diamond$.

Characteristics: One (if $n = 1$) or two variables, modal depth 0 (if $n = 1$) or $n^2 - n + 1$.

$s4\_md\_p(n) :\equiv$

p_1
$\vee \bigvee_{i=1,\ldots,n-1}(g(i,n,\neg p_1 \wedge \Diamond f(1,n,p_2)) \vee g(i,n,\neg p_1 \wedge \Diamond f(1,n,p_1)) \vee g(i,n,\neg p_2 \wedge \Diamond f(1,n,p_1)))$
$\vee g(n,n,\neg p_1)$

g as in 10.6.9.

$f(i,n,A) :\equiv h(i-1, \Box h(n-i,A))$

$$h(i,A) :\equiv \begin{cases} A & i = 0 \\ \Diamond h(i-1,A) \vee p_2 & \text{otherwise} \end{cases}$$

```
> for i := 1 to 6 do
    print(i, ": ", length(s4_md_p(i)), ", ", modaldepth(s4_md_p(i)), ", ", vars(s4_md_p(i)));
> s4_md_p(1);
> s4_md_p(2);
```

10.7.10 *s4_md_n*

Idea: Similar to 10.7.9, but simpler. The subformula $g(n, n, \neg p_1)$ is missing in order to make the formulas nonvalid.

Characteristics: One (if $n = 1$) or two variables, modal depth 0 (if $n = 1$) or $n^2 - n + 1$.

$s4\_md\_n(n) :\equiv p_1 \vee \bigvee_{i=1,\ldots,n-1}(g(i, n, \neg p_1 \wedge \Diamond f(1, n, p_1)))$

f, g, h as in 10.7.9.

```
> for i := 1 to 6 do
    print(i, ": ", length(s4_md_n(i)), ", ", modaldepth(s4_md_n(i)), ", ", vars(s4_md_n(i)));
> s4_md_n(1);
> s4_md_n(2);
```

10.7.11 *s4_path_p*

Idea, hiding: As in 10.6.11.

Characteristics: $6n$ variables, modal depth $n + 1$.

$$\begin{aligned} s4\_path\_p(n) :\equiv\ & (\Box\Box p_{11} \vee \Box\Box p_{10+path\_el(1,n)} \vee \Box\Box p_{13} \vee \Box\Box p_{15}) \\ & \vee \bigvee_{i=1,\ldots,n;j=1,\ldots,6}(left\_to\_right(i, j, n) \vee right\_to\_left(i, j, n)) \\ & \vee \Diamond(\Diamond\neg p_{10n+2} \vee \Diamond\neg p_{10n+4} \vee \Diamond\neg p_{10n+path\_el(n,n)} \vee \Diamond\neg p_{10n+6}) \end{aligned}$$

path_el, *left_to_right*, *right_to_left*, *delete*, *modg*, *lists2fml*, *lists2fml_back* as in 10.6.11.

```
> for i := 1 to 6 do
    print(i, ": ", length(s4_path_p(i)), ", ", modaldepth(s4_path_p(i)), ", ",
          vars(s4_path_p(i)));
> s4_path_p(1);
> s4_path_p(2);
```

10.7.12 *s4_path_n*

Idea, hiding: As in 10.6.12.

Characteristics: $6n$ variables, modal depth $n + 1$.

$$\begin{aligned} & s4\_path\_n(n) :\equiv \\ & \quad (\Box\Box p_{11} \vee \Box\Box p_{10+path\_el(1,n+1)} \vee \Box\Box p_{13} \vee \Box\Box p_{15}) \\ & \quad \vee \bigvee_{i=1,\ldots,n+1;j=1,\ldots,6}(left\_to\_right(i, j, n + 1) \vee right\_to\_left(i, j, n + 1)) \\ & \quad \vee \Diamond(\Diamond\neg p_{10(n+1)+2} \vee \Diamond\neg p_{10(n+1)+4} \vee \Diamond\neg p_{10(n+1)+path\_el((n+1),(n+1))} \vee \Diamond\neg p_{10(n+1)+6}) \end{aligned}$$

path_el, *left_to_right*, *right_to_left*, *delete*, *modg*, *lists2fml*, *lists2fml_back* as in 10.6.12.

```
> for i := 1 to 6 do
    print(i, ": ", length(s4_path_n(i)), ", ", modaldepth(s4_path_n(i)), ", ",
          vars(s4_path_n(i)));
> s4_path_n(1);
> s4_path_n(2);
```

10.7.13 *s4_ph_p*

Idea: The pigeonhole formulas. We assume $n < 100$.

Hiding: Some $\Box$ and $\Diamond$.

Characteristics: Essentially a CPC problem. $O(n^2)$ variables, modal depth 2 (if $n = 1$) or 3.

$s4\_ph\_p(n) :\equiv left(n) \rightarrow \Diamond right(n)$

$right(n) :\equiv \bigvee_{j=1,\ldots,n;i_1=1,\ldots,n+1;i_2=i_1+1,\ldots,n+1}(\Diamond(l(i_1, j) \wedge l(i_2, j)))$

left, l as in 10.5.13.

```
> for i := 1 to 6 do
    print(i, ": ", length(s4_ph_p(i)), ", ", modaldepth(s4_ph_p(i)), ", ", vars(s4_ph_p(i)));
> s4_ph_p(1);
> s4_ph_p(2);
```

10.7.14 *s4_ph_n*

Idea: The pigeonhole formulas, with one missing conjunct on the right hand side. We assume $n < 100$.

Hiding: Some $\Box$ and $\Diamond$.

Characteristics: Essentially a CPC problem. $O(n^2)$ variables, modal depth 2 (if $n = 1$) or 3.

$s4\_ph\_n(n) :\equiv left(n) \rightarrow \Diamond right(n)$

$right(n) :\equiv \bigvee_{j=1,\ldots,n;i_1=1,\ldots,n+1;i_2=i_1+1,\ldots,n+1}(\Diamond(l2(n,i_1,j) \wedge l2(n,i_2,j)))$

left, l as in 10.5.13.

$$l2(n,i,j) :\equiv \begin{cases} \neg l(i,j) & i = j, i = (2n) \text{ div } 3 + 1 \\ l(i,j) & \text{otherwise} \end{cases}$$

```
> for i := 1 to 6 do
    print(i, ": ", length(s4_ph_n(i)), ", ", modaldepth(s4_ph_n(i)), ", ", vars(s4_ph_n(i)));
> s4_ph_n(1);
> s4_ph_n(2);
```

10.7.15 *s4_s5_p*

Idea: A formula that is valid in S5 embedded in S4.

Hiding: Some superfluous subformulas.

Characteristics: $3n$ variables, modal depth $3n$.

$s4\_s5\_p(n) :\equiv \Box\Diamond(\Box \bigvee_{i=1,\ldots,3n-2}(p_i \leftrightarrow p_{i+1}) \vee \Box p_{3n} \vee f(1, 3n-1)) \vee \Box(\Diamond p_1 \rightarrow \neg p_{3n})$

$$f(i,n) :\equiv \begin{cases} \mathsf{false} & i = n \\ \Diamond(p_i \wedge \neg p_{i+1} \vee \neg p_i \wedge p_{i+1}) \vee \Box f(i+1,n) & \text{otherwise} \end{cases}$$

```
> for i := 1 to 6 do
    print(i, ": ", length(s4_s5_p(i)), ", ", modaldepth(s4_s5_p(i)), ", ", vars(s4_s5_p(i)));
> s4_s5_p(1);
> s4_s5_p(2);
```

10.7.16 *s4_s5_n*

Idea: A formula that is not valid in S5 embedded in S4.

Hiding: Some superfluous subformulas.

Characteristics: $6n$ variables, modal depth $6 - 4n$.

$s4\_s5\_n(n) :\equiv \Box\Diamond(\Box p_{6n} \vee f(1, 6n-5)) \vee \Box(\Diamond p_1 \rightarrow \neg p_{6n})$

f as in 10.7.15.

```
> for i := 1 to 6 do
    print(i, ": ", length(s4_s5_n(i)), ", ", modaldepth(s4_s5_n(i)), ", ", vars(s4_s5_n(i)));
> s4_s5_n(1);
> s4_s5_n(2);
```

10.7.17 *s4_t4p_p*

Idea, hiding: As in 10.5.17.

Characteristics: 4 variables, modal depth $2n+3$. Partially in negation normal form.

$$s4\_t4p\_p(n) :\equiv E(n) \vee \mathrm{nnf}(\neg C(2n-1)) \vee \Diamond p_4$$

$$C(i) :\equiv \begin{cases} (\Box(\Diamond\Box\Diamond p_0 \wedge p_0 \rightarrow \Box p_1) \wedge \Box\Diamond p_1)\{p_0 \wedge \Diamond\neg p_3/p_1\} & i = 0 \\ \mathrm{Dum}\{p_1/p_0\} \wedge \Box C(i-1) & \text{otherwise} \end{cases}$$

E as in 10.5.17.

```
> for i := 1 to 6 do
    print(i, ": ", length(s4_t4p_p(i)), ", ", modaldepth(s4_t4p_p(i)), ", ",
          vars(s4_t4p_p(i)));
> s4_t4p_p(1);
> s4_t4p_p(2);
```

10.7.18 *s4_t4p_n*

Idea, hiding: As in 10.5.18.

Characteristics: 4 variables, modal depth $4n+3$. Partially in negation normal form.

$$s4\_t4p\_n(n) :\equiv E(2n-1) \vee \mathrm{nnf}(\neg C(4n-1)) \vee \Diamond p_4$$

$$C(i) :\equiv \begin{cases} ((\Box\Diamond p_0 \rightarrow \Diamond p_1) \wedge \Box(\Diamond\Box\Diamond p_0 \wedge p_0 \rightarrow \Box p_1))\{p_0 \wedge \Diamond\neg p_3/p_1\} & i = 0 \\ \mathrm{Dum}\{p_1/p_0\} \wedge \Box C(i-1) & \text{otherwise} \end{cases}$$

E as in 10.5.17.

```
> for i := 1 to 6 do
    print(i, ": ", length(s4_t4p_n(i)), ", ", modaldepth(s4_t4p_n(i)), ", ",
          vars(s4_t4p_n(i)));
> s4_t4p_n(1);
> s4_t4p_n(2);
```

10.8 LWB

Prover: LWB, version 1.0

Availability, additional facilities of the prover: See chapter 8.

Hardware: Sun SPARCstation 5, main memory: 80 MB, 1 CPU (70 MHz microSPARC II)

Timing: The timing includes the parsing of the formulas and the construction of the corresponding data structure. The files loaded by the LWB have the following form:

```
load(k);
timestart; provable(box p0 -> box box p0); timestop;
quit;
```

Results:

class	$n_{\mathrm{p},i}$	class	$n_{\mathrm{n},i}$
k_branch_p	6	*k_branch_n*	7
k_d4_p	8	*k_d4_n*	6
k_dum_p	13	*k_dum_n*	19
k_grz_p	7	*k_grz_n*	13
k_lin_p	11	*k_lin_n*	8
k_path_p	12	*k_path_n*	10
k_ph_p	4	*k_ph_n*	8
k_poly_p	8	*k_poly_n*	11
k_t4p_p	8	*k_t4p_n*	7

class	$n_{\mathrm{p},i}$	class	$n_{\mathrm{n},i}$
kt_45_p	5	*kt_45_n*	4
kt_branch_p	5	*kt_branch_n*	6
kt_dum_p	5	*kt_dum_n*	10
kt_grz_p	6	*kt_grz_n*	> 20
kt_md_p	5	*kt_md_n*	5
kt_path_p	10	*kt_path_n*	9
kt_ph_p	4	*kt_ph_n*	8
kt_poly_p	14	*kt_poly_n*	2
kt_t4p_p	5	*kt_t4p_n*	7

class	$n_{\mathrm{p},i}$	class	$n_{\mathrm{n},i}$
s4_45_p	3	*s4_45_n*	5
s4_branch_p	11	*s4_branch_n*	7
s4_grz_p	9	*s4_grz_n*	> 20
s4_ipc_p	8	*s4_ipc_n*	7
s4_md_p	8	*s4_md_n*	6
s4_path_p	8	*s4_path_n*	6
s4_ph_p	4	*s4_ph_n*	8
s4_s5_p	4	*s4_s5_n*	9
s4_t4p_p	9	*s4_t4p_n*	12

In order to make absolutely clear how we obtained the numbers in the tables above from the run times, we give the run times for some of the formulas in the classes *k_branch_p* and *k_branch_n*. *k_branch_p*(6) and *k_branch_n*(7) are the last formulas that could be decided in less than 100 seconds; therefore we enter the numbers 6 and 7 in the first line of the table for K.

formula	run time (in seconds)
k_branch_p(1)	0.02
k_branch_p(2)	0.06
k_branch_p(3)	0.28
k_branch_p(4)	1.80
k_branch_p(5)	11.89
k_branch_p(6)	74.64
k_branch_p(7)	503.94

formula	run time (in seconds)
k_branch_n(1)	0.02
k_branch_n(2)	0.05
k_branch_n(3)	0.15
k_branch_n(4)	0.51
k_branch_n(5)	2.17
k_branch_n(6)	9.69
k_branch_n(7)	43.42
k_branch_n(8)	203.81

10.9 AVAILABILITY OF THE FORMULAS

The first 15 formulas of each class as well as the LWB programs that generated these formulas are available via the LWB home page `http://lwbwww.unibe.ch:8080/LWBinfo.html`.

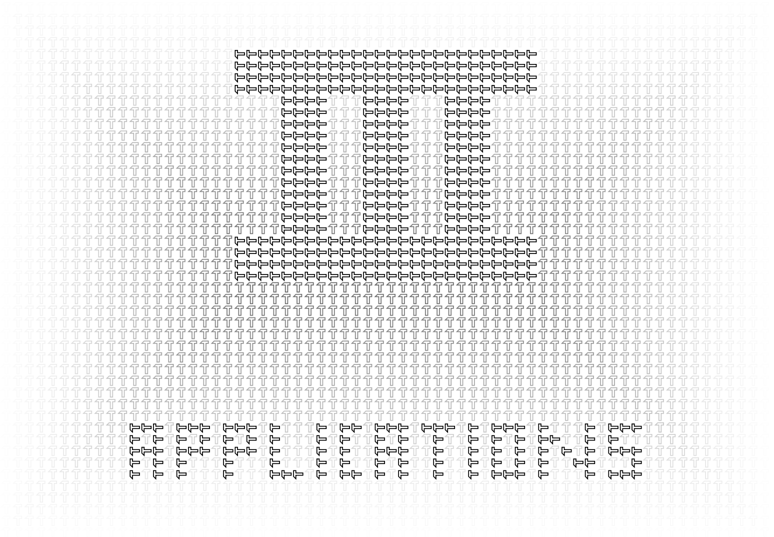
For each class of the three logics K, KT, S4 there is a compressed ASCII file (inside a tar file) that contains the first 21 formulas. The files contain a header line, a line with the word `begin` , then one formula on each line (with the corresponding number in front), and finally a line with the word `end` . The formulas are written in infix notation. The connectives are `&` for $\wedge$, `v` for $\vee$, `->` for $\rightarrow$, `<->` for $\leftrightarrow$, and a tilde for $\neg$. No brackets are omitted in the formulas in order to make the conversion into other formats easier.

It can happen that for some classes more than 21 formulas are needed. Then there are several possibilities:

- Get the LWB procedures we used to generate the formulas, install the LWB, and generate the required formulas.
- Write a procedure that generates the formulas in this class according to the definitions in the sections 10.5, 10.6, 10.7. Please make sure that you generate the same formulas as we do by comparing the first 21 formulas.
- Get the LWB procedures and use the LWB via WWW (choose the item `run a session` on the LWB home page). You have to replace the `read` statements in the files with the corresponding files for this purpose. Use `set("bracketmode", "full");` to obtain all the brackets.

10.10 SUMMARY

We have investigated the requirements for benchmark methods in the area of decision procedures for logics. Beginning with these requirements we have developed a benchmark method for the logics K, KT, S4. This method should be applicable to many other logics.


III
APPLICATIONS

11

RELATIONS BETWEEN NORMAL MODAL LOGICS

‘We can send a microphotograph in an ordinary letter. You stick it on as a full stop and they float it in the water until the dot comes unstuck. I suppose you do write letters home sometimes. Business letters . . . ?’
‘I send those to New York.’
‘Friends and relations?’

G. Greene. Our man in Havanna.

11.1 INTRODUCTION

In this thesis, we investigate just three propositional normal modal logics in detail, namely K, KT, and $\mathsf{S4}$. However, many others exist. Propositional normal modal logics are essentially extensions of the logic K. We can define them via Hilbert-style calculi: We take the calculus $\mathsf{K}^{\mathcal{H}}$ (see definition 2.2.1) and add new axioms. The Hilbert-style calculus $\mathsf{KT}^{\mathcal{H}}$, for example, is obtained from $\mathsf{K}^{\mathcal{H}}$ by adding a formula scheme that corresponds to the formula T as an axiom (cp. definition 2.4.2), and the Hilbert-style calculus $\mathsf{S4}^{\mathcal{H}}$ is obtained from $\mathsf{K}^{\mathcal{H}}$ by adding a formula schemes that correspond to the formulas T and 4 as axioms (cp. definition 2.6.2). Note that we cannot use $\mathsf{K} + T$ for this purpose, since there the theory T must be finite, whereas adding an axiom to K means that all its (i.e. in general infinitely many) instances are added.

In this chapter we investigate the relations between the ‘usual’ propositional normal modal logics. More exactly, we compare, using the $\subseteq$ relation, the sets of formulas that are provable in these logics. An obvious example is $\{A \mid \mathsf{K}^{\mathcal{H}} \vdash A\} \subseteq \{A \mid \mathsf{KT}^{\mathcal{H}} \vdash A\}$.

11.2 THEORY

11.2.1 DEFINITION propositional normal modal logics

A propositional modal logic L is normal if the set of formulas that are provable in L can be defined with a Hilbert-style calculus that contains the axioms and rules of $\mathsf{K}^{\mathcal{H}}$ plus finitely many additional axioms. If $A_1, \ldots, A_n$ are formulas, then we write $\mathsf{KA}_1 \ldots \mathsf{A}_n{}^{\mathcal{H}}$ for the Hilbert-style calculus that consists of $\mathsf{K}^{\mathcal{H}}$ plus formula schemes that correspond to $A_1, \ldots, A_n$ as axioms.

11.2.2 REMARK $\mathsf{KT4}^{\mathcal{H}}$ and $\mathsf{S4}^{\mathcal{H}}$

Note that the calculus $\mathsf{S4}^{\mathcal{H}}$ is called $\mathsf{KT4}^{\mathcal{H}}$ in this chapter, and that $\mathsf{S5}^{\mathcal{H}}$ is called $\mathsf{KT45}^{\mathcal{H}}$.

11.2.3 DEFINITION boxed instance of a formula

A boxed instance of a formula A is a formula of the form $\Box^n A\{C_0/p_0, \ldots, C_m/p_m\}$.

Example

Let A be the formula $\Box p_0 \to \Diamond p_1$. Then $\Box^2(\Box p_0 \to \Diamond p_1)\{p_0 \vee \Diamond p_2/p_0\} \equiv \Box\Box(\Box(p_0 \vee \Diamond p_2) \to \Diamond p_1)$ is a boxed instance of A.

11.2.4 DEFINITION provability

$\mathsf{KA}_1 \ldots \mathsf{A}_n{}^{\mathcal{H}} \vdash C$ iff there exist $m_1, \ldots, m_n \in \mathbb{N}$ such that:

- $\forall i \in \{1, \ldots, n\} : \forall j \in \{1, \ldots, m_i\} : A_{i,j}$ is a boxed instance of the formula A_i
- $\mathsf{K}^{\mathcal{H}} \vdash \bigwedge_{i \in \{1,\ldots,n\}, j \in \{1,\ldots,m_i\}} (A_{i,j}) \to B$

11.2.5 THEOREM equality

Let $\mathsf{KA}_1 \ldots \mathsf{A}_{n_1}{}^{\mathcal{H}}$ and $\mathsf{KC}_1 \ldots \mathsf{C}_{n_2}{}^{\mathcal{H}}$ be two Hilbert-style calculi of two propositional normal modal logics. Then $\{D \mid \mathsf{KA}_1 \ldots \mathsf{A}_{n_1}{}^{\mathcal{H}} \vdash D\} = \{D \mid \mathsf{KC}_1 \ldots \mathsf{C}_{n_2}{}^{\mathcal{H}} \vdash D\}$ if we have:

- $\forall i \in \{1, \ldots, n_1\} : \mathsf{KC}_1 \ldots \mathsf{C}_{n_2}{}^{\mathcal{H}} \vdash A_i$
- $\forall i \in \{1, \ldots, n_2\} : \mathsf{KA}_1 \ldots \mathsf{A}_{n_1}{}^{\mathcal{H}} \vdash C_i$

11.3 RESULTS

11.3.1 REMARK how to read the tables

In this section, a row in a table consists of a list of formulas $A_1, \ldots, A_n$, a formula C, and a list of formulas $A'_1, \ldots, A'_m$, where $A'_1, \ldots, A'_m$ are boxed instances of the formulas $A_1, \ldots, A_n$.

Such a row means:

- $\mathsf{KA}_1 \ldots \mathsf{A}_n{}^{\mathcal{H}} \vdash C$
- $\mathsf{K}^{\mathcal{H}} \vdash A'_1 \wedge \ldots \wedge A'_m \to C$

Example

The row

T, 5	4	T$\{\Diamond\neg p_0/p_0\}$, 5$\{\Box p_0/p_0\}$, $\Box$5$\{\neg p_0/p_0\}$

means that $\mathsf{K}^{\mathcal{H}} \vdash \mathrm{T}\{\Diamond\neg p_0/p_0\} \wedge 5\{\Box p_0/p_0\} \wedge \Box 5\{\neg p_0/p_0\} \to 4$. Thus $\mathsf{KT5}^{\mathcal{H}} \vdash 4$, and T$\{\Diamond\neg p_0/p_0\}$, 5$\{\Box p_0/p_0\}$, $\Box$5$\{\neg p_0/p_0\}$ are the required boxed instances of the formulas T and 5.

11.3.2 REMARK searching for boxed instances

Using a decision procedure for K, we can search for boxed instances in a systematic way, i.e. we immediately obtain semi-decision procedures for all propositional normal modal logics.

Without any guidance, only the most trivial examples can be solved. A first optimisation is to start with the decision procedure for KT and S4 whenever possible. Consider for example the case where we wanted to prove that $\mathsf{KT4R}^{\mathcal{H}} \vdash \mathrm{Dum}$. We first used the decision procedure for S4 in order to find the required boxed instances of R. Only afterwards did we look for the instances of T and 4. In fact, these instances can be easily computed (cp. theorem 7.4.3).

We soon observed that in many cases it is sufficient to consider just the boxed instances with no or just one $\Box$ in front. Often it is also sufficient to substitute the variables only by subformulas of the formula we want to prove. Unfortunately, there are counterexamples, for example already in the simple case $\mathsf{KW}^{\mathcal{H}} \vdash 4$.

Assume for example that we consider a case where only subformulas of the formula we want to prove are required to build the boxed instances and where at most two $\Box$ are required. Then in theory, a fully automatic search with a program in the LWB programming language would be possible. However, sometimes a lot of time is used to decide whether a certain formula is provable in a logic L even if 'similar' formulas were easy, i.e. the automatic search gets stuck. To overcome this problem, the program would have to stop the proof search for a formula after a while and to continue with the next one. This is not possible with the LWB 1.0, but is part of a forthcoming version.

Many of the boxed instances given in the tables in this section have been found during a search with the LWB. However, this has only been possible because of the help of the human modal logic experts Rajeev Goré and Wolfgang Heinle, who made suggestions on what the boxed instances could look like.

Example

We try to prove $\mathsf{KT5}^{\mathcal{H}} \vdash 4$. First we work in KT and try to find the boxed instances of the formula 5. To build those boxed instances, we use a list of the most simple formulas with one variable and assume that one $\Box$ is sufficient. (Note that in KT, the formula $\Box A \to C$ is valid iff the formula $A \wedge \Box A \to C$ is valid.) We find a solution if we take two boxed instances.

```
> load(kt);
> A4 :== box p0 -> box box p0;
  A5 :== dia p0 -> box dia p0;
> l :== [p0, ~p0, box p0, box ~p0, dia p0, dia ~p0];
> foreach x1 in l do
    if provable(box A5{x1/p0} -> A4) then print(x1);
> foreach x1 in l do
    foreach x2 in l do
      if kt::provable(box A5{x1/p0} & box A5{x2/p0} -> A4)
      then print(x1, ", ", x2);
```

We check whether the $\Box$ are necessary and find that we can omit one of them.

```
> kt::provable(A5{~p0/p0} & A5{box p0/p0} -> A4);
> kt::provable(A5{~p0/p0} & box A5{box p0/p0} -> A4);
> kt::provable(box A5{~p0/p0} & A5{box p0/p0} -> A4);
```

Now we still have to find the instances of T. We make use of theorem 7.4.3. The $\Diamond$ formulas in $\mathrm{nnf}(5\{\neg p_0/p_0\} \wedge \Box 5\{\neg p_0/p_0\} \wedge \mathrm{A5}\{\Box p_0/p_0\} \rightarrow 4)$ are $\{\Diamond\neg p_0, \Diamond\Box p_0, \Diamond(\Diamond\neg p_0 \wedge \Diamond\Box p_0), \Diamond\Box\Diamond\neg p_0\}$. Therefore it is sufficient to consider the instances T, $\mathrm{T}\{\Diamond\neg p_0/p_0\}$, $\mathrm{T}\{\Box p_0 \vee \Box\Diamond\neg p_0/p_0\}$, and $\mathrm{T}\{\Diamond\Box p_0/p_0\}$. We first use theories to find a minimal set of these instances and then check how many $\Box$ are needed.

Two remarks: In practice, it is often better not to work with theories, but to assume that at most one or two $\Box$ are needed, because it is easier to decide the validity of these formulas. And of course trial and error would be a more efficient solution in this simple case, but look for example at the formulas in theorem 11.3.9 . . .

Note that we have to use both $\Box 5\{\neg p_0/p_0\}$ and $5\{\neg p_0/p_0\}$, since $A \wedge \Box A \rightarrow C$ and $\Box A \rightarrow C$ are not equivalent in K.

```
> load(k);
> T :== box p0 -> p0;
> l :== [T, T{dia ~p0/p0}, T{box p0 v box dia ~p0/p0}, T{dia box p0/p0}];
> k::provable(A5{~p0/p0} & box A5{~p0/p0} & A5{box p0/p0} -> A4, [l[1]]);
> k::provable(A5{~p0/p0} & box A5{~p0/p0} & A5{box p0/p0} -> A4, [l[1],l[2]]);
> k::provable(A5{~p0/p0} & box A5{~p0/p0} & A5{box p0/p0} -> A4, [l[2]]);
> k::provable(l[2] & box l[2] & A5{~p0/p0} & box A5{~p0/p0} & A5{box p0/p0} -> A4);
```

Finally we check whether we need both $5\{\neg p_0/p_0\}$ and $\Box 5\{\neg p_0/p_0\}$ or whether $\Box 5\{\neg p_0/p_0\}$ is sufficient, and analogously for $\mathrm{T}\{\Diamond\neg p_0/p_0\}$.

```
> k::provable(T{dia ~p0/p0} & A5{~p0/p0} & box A5{~p0/p0} & A5{box p0/p0} -> A4);
> k::provable(T{dia ~p0/p0} & box A5{~p0/p0} & A5{box p0/p0} -> A4);
```

Thus we end up with the boxed instances shown in theorem 11.3.4. Of course the search is not always so simple, especially if the formulas are longer or contain two variables.

11.3.3 REMARK checking the results

Probably most of the results in the tables below can be found somewhere in the literature. However, this literature is sometimes almost inaccessible (for example [Seg71]) or uses a nowadays unusual notation (see for example [Sob64], [Sob70]). In addition, it is rather difficult to check these results, since a variety of methods has been used to prove them.

The tables below do not only summarise the most important relations. Thanks to the boxed instances in the third column, we can check each result by deciding the validity of a formula in K. This can be done using a theorem prover, but it is also possible to do it by hand, using for example backward proof search in $\mathsf{K}^{\mathcal{S},2}$ (although it can be annoying in some of the more complicated cases). No additional knowledge (for example about semantics, algebra, . . .) is required.

11.3.4 THEOREM relations concerning $\mathrm{D}, \mathrm{T}, 4, 5, \mathrm{B}$

$\mathsf{KA}_1 \ldots \mathsf{A}_k \vdash B$		
$A_1, \ldots, A_k$	B	
D	D_2	D
D_2	D	D_2
4	4_M	4
4_M	4	$4_\mathrm{M}\{\neg\Box p_0/p_1\}$
5	5_M	5
5_M	5	$5_\mathrm{M}\{\neg\Diamond p_0/p_1\}$
B	B_M	B
B_M	B	$\mathrm{B}_\mathrm{M}\{\neg\Diamond p_0/p_1\}$
T	D	$\mathrm{T}\{\mathsf{false}/p_0\}$
T, 5	4	$\mathrm{T}\{\Diamond\neg p_0/p_0\}$, $5\{\Box p_0/p_0\}$, $\Box 5\{\neg p_0/p_0\}$
T, 5	B	$\mathrm{T}\{\neg p_0/p_0\}$, 5
4, B	5	$\Box 4\{\neg p_0/p_0\}$, $\mathrm{B}\{\Diamond p_0/p_0\}$
5, B	4	$\Box 5\{\neg p_0/p_0\}$, $\mathrm{B}\{\Box p_0/p_0\}$
D, 4, B	T	D, 4, $\mathrm{B}\{\neg p_0/p_0\}$

See for example [Che80] for a complete diagram of the logics that are obtained by adding combinations of T, D, 4, 5, B as axioms to K.

11.3.5 THEOREM relations concerning $\mathrm{G}, \mathrm{L}, \mathrm{H}$

$\mathsf{KA}_1 \ldots \mathsf{A}_k \vdash B$		
$A_1, \ldots, A_k$	B	
G	G_0	$\mathrm{G}\{p_1/p_0\}$
D, G_0	G	$\Box\mathrm{D}$, $\mathrm{G}_0\{\Box p_0/p_0, p_0/p_1\}$
L	H	L
H	L	$\mathrm{H}\{p_0 \wedge \Box p_0 \to p_1/p_0, p_1 \wedge \Box p_1 \to p_0/p_1\}$
L^+	H^+	L^+
H^+	L^+	$\mathrm{H}^+\{\Box p_0 \to p_1/p_0, \Box p_1 \to p_0/p_1\}$
L^+	L	L^+
T, L	L^+	$\Box\mathrm{T}$, $\Box\mathrm{T}\{p_1/p_0\}$, L
$\mathrm{T}, \mathrm{L}^{++}$	L^+	$\Box\mathrm{T}$, $\Box\mathrm{T}\{p_1/p_0\}$, L^{++}
$4, \mathrm{L}$	L^{++}	$4\{p_0 \wedge \Box p_0 \to p_1/p_0\}$, $4\{p_1 \wedge \Box p_1 \to p_0/p_0\}$, $\Box 4$, $\Box 4\{p_1/p_0\}$, L
5	L^{++}	$5\{\Box p_0/p_0\}$, $\Box 5\{\neg p_0/p_0\}$
$\mathrm{D}, \mathrm{L}^{++}$	G	$\Box\mathrm{D}\{\neg p_0/p_0\}$, $\mathrm{L}^{++}\{\neg p_0/p_1\}$
5	G	5, $5\{\neg p/p_0\}$

11.3.6 THEOREM relations concerning M

$\mathsf{KA}_1 \dots \mathsf{A}_k \vdash B$		
$A_1, \dots, A_k$	B	
$\mathrm{T}, 4, \mathrm{M}$	M_2	$\mathrm{T}\{\mathrm{M}/p_0\}$, $\Box 4$, $\Box \mathrm{M}$
$\mathrm{T}, 4, \mathrm{M}_2$	M	$\mathrm{T}\{\mathrm{M}_2/p_0\}$, $4\{\Diamond\neg p_0/p_0\}$, M_2
$4, \mathrm{M}$	M_3	$4\{\Diamond p_1 \wedge (\neg p_0 \vee \neg p_1)/p_0\}$, M
M_3	M	$\mathrm{M}_3\{\neg p_0/p_1\}$
M	D	M
$4, \mathrm{M}$	Pt	$4\{(p_0 \vee \Diamond p_0) \wedge (\neg p_0 \vee \Diamond\neg p_0)/p_0\}$, $\Box 4\{p_0 \wedge \Diamond\neg p_0/p_0\}$, $\Box 4\{\neg p_0 \wedge \Diamond p_0/p_0\}$, M
Pt	M	Pt

Variants of M have been investigated in [Sob64]. Pt was introduced in [Hei95].

11.3.7 THEOREM relations concerning W

$\mathsf{KA}_1 \dots \mathsf{A}_k \vdash B$		
$A_1, \dots, A_k$	B	
W	4	$\mathrm{W}\{p_0 \wedge \Box p_0/p_0\}$
W	W_0	$\mathrm{W}\{\mathsf{false}/p_0\}$
W	Z	W
W_0, Z	W	W_0, Z
W	Grz_1	$\mathrm{W}\{p_0 \wedge \Box(\Box p_0 \to p_0)/p_0\}$, $\Box\Box \mathrm{W}$
$4, \mathrm{Z}$	Dum_1	$\Box 4, \mathrm{Z}$

11.3.8 THEOREM miscellaneous relations

$\mathsf{KA}_1 \dots \mathsf{A}_k \vdash B$		
$A_1, \dots, A_k$	B	
R	L^+	$\mathrm{R}\{\Box p_0 \to p_1/p_0\}$
T, R	Zem	$\mathrm{T}\{\Diamond\Box p_0/p_0\}$, R
$\mathrm{T}, 4, \mathrm{L}^+, \mathrm{Zem}$	R	$\Box\Box\mathrm{T}\{4/p_0\}$, $\Box\Box\mathrm{T}\{\neg(\Box\Diamond\Box p_0 \wedge \Box p_0)/p_0\}$, $\Box 4$, $\mathrm{L}^+\{\Diamond\Box p_0/p_0, \neg\Box p_0/p_1\}$, Zem
$\mathrm{T}, 4, \mathrm{P}$	M	$\mathrm{T}\{\Diamond p_0/p_0\}$, $\Box\mathrm{T}\{\neg\Box\Diamond p_0/p_0\}$, $4\{\Diamond p_0/p_0\}$, $\Box\mathrm{P}$
T, P	R	$\Box\Box\mathrm{T}\{\neg p_0/p_0\}$, P
$4, \mathrm{M}, \mathrm{R}$	P	$4\{\neg\Box p_0/p_0\}$, $\Box\mathrm{M}$, R
T, P	H_s	$\Box\Box\Box\mathrm{T}$, P, $\Box\mathrm{P}\{\neg p_0/p_0\}$
$\mathrm{T}, 4, \mathrm{R}$	Dum	$\mathrm{T}\{\Box(p_0 \to \Box p_0) \to p_0/p_0\}$, $\Box 4\{p_0 \wedge (\Box(p_0 \to \Box p_0) \to p_0)/p_0\}$, $\mathrm{R}\{p_0 \to \Box p_0/p_0\}$

Most of these relations were already proved in [Sob70].

11.3.9 THEOREM relations concerning Dum, Grz

Figure 11.a contains the instances concerning the formulas Dum, Grz and their variants. A few results concerning the equivalence of variants of Grz and Dum can be found in [Sob70]. In [Seg71] it is proved, using semantic means, that our variants of Dum are "deductively equivalent".

Note that about 20 years ago, the result $\mathsf{KGrz}^{\mathcal{H}} \vdash 4$ (near the bottom of the table) was still worth a paper (see [vB87]).

11.3.10 REMARK a missing row

In an exercise in [Che80] it is stated that $\mathsf{KL}^{+\mathcal{H}} \vdash \mathrm{G}$. We have not been able to find instances of L^{+} in order to prove this relationship.

11.3.11 REMARK diagrams, formula names

See [GHH97] for diagrams that summarise the relationships listed in the tables below. There you can also find a table that helps to translate our formula names into other names used in the literature.

11.4 LWB SESSION

The following LWB session corresponds to the tables given in the previous section.

```
> load(k);
> D :== box p0 -> dia p0;
  D2 :== dia true;
  T :== box p0 -> p0;
  A4 :== box p0 -> box box p0;
  A4m :== box p0 & dia p1 -> dia(box p0 & p1);
  A5 :== dia p0 -> box dia p0;
  A5m :== dia p0 & dia p1 -> dia(dia p0 & p1);
  B :== p0 -> box dia p0;
  Bm :== p0 & dia p1 -> dia(dia p0 & p1);
> G :== dia box p0 -> box dia p0;
  G0 :== dia(p0 & box p1) -> box(p0 v dia p1);
  H :== box(p0 v p1) & box(box p0 v p1) & box(p0 v box p1) -> box p0 v box p1;
  Hp :== box(box p0 v p1) & box(p0 v box p1) -> box p0 v box p1;
  L :== box(p0 & box p0 -> p1) v box(p1 & box p1 -> p0);
  Lp :== box(box p0 -> p1) v box(box p1 -> p0);
  Lpp :== box(box p0 -> box p1) v box(box p1 -> box p0);
> M :== box dia p0 -> dia box p0;
  Pt :== box(p0 v dia p0) -> dia(p0 & box p0);
> W :== box(box p0 -> p0) -> box p0;
  W0 :== box dia true -> box false;
  Z :== box(box p0 -> p0) -> (dia box p0 -> box p0);
> Dum :== box(box(p0 -> box p0) -> p0) -> (dia box p0 -> p0);
  Dum1 :== box(box(p0 -> box p0) -> p0) -> (dia box p0 -> box p0);
  Grz :== box(box(p0 -> box p0) -> p0) -> p0;
  Grz1 :== box(box(p0 -> box p0) -> p0) -> box p0;
> F :== (dia box p0 -> p1) v box(box p1 -> p0);
```

$\mathsf{KA}_1 \ldots \mathsf{A}_k \vdash B$		
$A_1, \ldots, A_k$	B	
$4, \mathrm{Dum}$	Dum_1	$4\{\Box(p_0 \to \Box p_0) \to p_0/p_0\}$, $\Box 4$, Dum, $\mathrm{Dum}\{p_0 \to \Box p_0/p_0\}$
$\mathrm{T}, \mathrm{Dum}_1$	Dum	T, Dum_1
T, Dum	Dum_2	$\Box\mathrm{T}$, Dum
$\mathrm{T}, \mathrm{Dum}_2$	Dum	T, $\Box\mathrm{T}\{p_0 \to \Box p_0/p_0\}$, Dum_2
$\mathrm{T}, 4, \mathrm{Dum}$	Dum_3	$\Box\mathrm{T}$, $4\{\Box(p_0 \to \Box p_0) \to p_0/p_0\}$, $\Box 4$, Dum, $\mathrm{Dum}\{p_0 \to \Box p_0/p_0\}$
$\mathrm{T}, \mathrm{Dum}_3$	Dum	T, $\Box\mathrm{T}\{p_0 \to \Box p_0/p_0\}$, Dum_3
Dum	Dum_4	Dum
$\mathrm{T}, \mathrm{Dum}_4$	Dum	T, Dum_4
Grz	Grz_1	$\mathrm{Grz}\{(\Box(p_0 \to \Box p_0) \to p_0) \wedge 4\{\Box(p_0 \to \Box p_0) \to p_0/p_0\}/p_0\}$, $\Box\mathrm{Grz}$
$\mathrm{T}, \mathrm{Grz}_1$	Grz	T, $\mathrm{T}\{\mathrm{Grz}_1/p_0\}$, $\Box\mathrm{Grz}_1$
Grz	Grz_2	Grz, $\Box\mathrm{Grz}$
$\mathrm{T}, \mathrm{Grz}_2$	Grz	$\Box\mathrm{T}\{p_0 \to \Box p_0/p_0\}$, $\mathrm{T}\{\neg\Box(\Box(p_0 \to \Box p_0) \to p_0)/p_0\}$, Grz_2
Grz	Grz_3	$\mathrm{Grz}\{(\Box(p_0 \to \Box p_0) \to \Box p_0) \wedge \mathrm{Grz}$ $\wedge\, 4\{(\Box(p_0 \to \Box p_0) \to \Box p_0) \wedge \mathrm{Grz}/p_0\}/p_0\}$, $\Box\mathrm{Grz}$
$\mathrm{T}, \mathrm{Grz}_3$	Grz	T, $\Box\mathrm{T}\{p_0 \to \Box p_0/p_0\}$, Grz_3
Grz	Grz_4	Grz, $\mathrm{Grz}\{(\Box(p_0 \to \Box p_0) \to p_0) \wedge 4\{\Box(p_0 \to \Box p_0) \to p_0/p_0\}/p_0\}$
$\mathrm{T}, \mathrm{Grz}_4$	Grz	T, Grz_4
Grz	Grz_5	$\mathrm{Grz}\{(\Box(p_0 \to \Box p_1) \to \Box p_1) \wedge 4\{\Box(p_0 \to \Box p_1) \to \Box p_1/p_0\}/p_0\}$, $\mathrm{Grz}\{(\Box((p_0 \to \Box p_1) \to \Box(p_0 \to \Box p_1)) \to (p_0 \to \Box p_1))$ $\wedge\, 4\{\Box((p_0 \to \Box p_1) \to \Box(p_0 \to \Box p_1)) \to (p_0 \to \Box p_1)/p_0\}/p_0\}$, $\Box\mathrm{Grz}\{p_0 \to \Box p_1/p_0\}$
$\mathrm{T}, 4, \mathrm{Grz}_5$	Grz	$\mathrm{T}\{\Box(p_0 \to \Box p_0) \to p_0/p_0\}$, $\Box\Box\mathrm{T}\{p_0 \to \Box p_0/p_0\}$, $4\{\Box(p_0 \to \Box p_0) \to p_0/p_0\}$, $\Box 4$, $\mathrm{Grz}_5\{p_0 \to \Box p_0/p_1\}$
Grz	T	Grz
Grz	4	$\mathrm{Grz}\{p_0 \wedge 4/p_0\}$
Grz	M	$\mathrm{Grz}\{\Diamond\neg p_0/p_0\}$, $\mathrm{Grz}\{\Diamond\neg p_0 \wedge 4\{\Diamond\neg p_0/p_0\}/p_0\}$, $\mathrm{Grz}\{\neg\Box(p_0 \to \Box p_0) \wedge 4\{\neg\Box(p_0 \to \Box p_0)/p_0\}/p_0\}$, $\Box\mathrm{Grz}$
Grz	Dum	Grz
Grz_1	Dum_1	Grz_1
$\mathrm{T}, \mathrm{M}, \mathrm{Dum}$	Grz	$\Box\mathrm{T}\{\neg p_0/p_0\}$, M, Dum

Figure 11.a: The instances from theorem 11.3.9.

```
  Hs :== p0 -> box(dia p0 -> p0);
  P :== dia box dia p0 -> (p0 -> box p0);
  R :== dia box p0 -> (p0 -> box p0);
  X :== box box p0 -> box p0;
  Zem :== box dia box p0 -> (p0 -> box p0);
> M2 :== dia box(p0 -> box p0);
  M3 :== box dia p0 & box dia p1 -> dia(p0 & p1);
> Dum2 :== box(box(p0 -> box p0) -> box p0) -> (dia box p0 -> p0);
  Dum3 :== box(box(p0 -> box p0) -> box p0) -> (dia box p0 -> box p0);
  Dum4 :== box(box(p0 -> box p0) -> p0)     -> (dia box p0 -> p0 v box p0);
> Grz2 :== box(box(p0 -> box p0) -> box p0) -> p0;
  Grz3 :== box(box(p0 -> box p0) -> box p0) -> box p0;
  Grz4 :== box(box(p0 -> box p0) -> p0)     -> p0 v box p0;
  Grz5 :== box(box(p0 -> box p1) -> box p1) & box(box(~p0 -> box p1) -> box p1)
           -> box p1;
```

Relations concerning D, T, 4, 5, B:

```
> k::provable( D  ->  D2 );
> k::provable( D2  ->  D );
> k::provable( A4  ->  A4m );
> k::provable( A4m{~ box p0/p1}  ->  A4 );
> k::provable( A5  ->  A5m );
> k::provable( A5m{~ dia p0/p1}  ->  A5 );
> k::provable( B  ->  Bm );
> k::provable( Bm{~ dia p0/p1}  ->  B );
> k::provable( T{false/p0}  ->  D );
> k::provable( T{dia ~p0/p0} & A5{box p0/p0} & box A5{~p0/p0}  ->  A4 );
> k::provable( T{~p0/p0} & A5  ->  B );
> k::provable( box A4{~p0/p0} & B{dia p0/p0}  ->  A5 );
> k::provable( box A5{~p0/p0} & B{box p0/p0}  ->  A4 );
> k::provable( D & A4 & B{~p0/p0}  ->  T );
```

Relations concerning G, G_0, H, H^+ L, L^+, L^{++}:

```
> k::provable( G{p1/p0} ->  G0 );
> k::provable( box D & G0{box p0/p0, p0/p1}  ->  G );
> k::provable( L  ->  H );
> k::provable( H{p0 & box p0 -> p1/p0, p1 & box p1 -> p0/p1}  ->  L );
> k::provable( Lp  ->  Hp );
> k::provable( Hp{box p0 -> p1/p0, box p1 -> p0/p1}  ->  Lp );
> k::provable( Lp  ->  L );
> k::provable( box T & box T{p1/p0}  & L  ->  Lp );
> k::provable( box T & box T{p1/p0} & Lpp  ->  Lp );
> k::provable(   A4{p0 & box p0 -> p1/p0} & A4{p1 & box p1 -> p0/p0}
               & box A4 & box A4{p1/p0}
               & L  ->  Lpp );
> k::provable( A5{box p0/p0} & box A5{~p0/p0}  ->  Lpp );
> k::provable( box D{~p0/p0} & Lpp{~p0/p1}  ->  G );
> k::provable( A5 & A5{~p0/p0}  ->  G );
```

Relations concerning M, Pt:

```
> k::provable( T{M/p0} & box A4 & box M  ->  M2 );
> k::provable( T{M2/p0} & A4{dia ~p0/p0} & M2  ->  M );
> k::provable( A4{dia p1 & (~p0 v ~p1)/p0} & M  ->  M3 );
> k::provable( M3{~p0/p1}  ->  M );
> k::provable( M  ->  D );
> k::provable(      A4{      (p0 v dia p0) & (~p0 v dia ~p0)/p0}
               & box A4{ p0 & (p0 v dia p0) & (~p0 v dia ~p0)/p0}
               & box A4{~p0 & (p0 v dia p0) & (~p0 v dia ~p0)/p0}
               & M
               ->  Pt );
> k::provable( Pt  ->  M );
```

Relations concerning W, W_0, Z:

```
> k::provable( W{p0 & box p0/p0}  ->  A4 );
> k::provable( W{false/p0}  ->  W0 );
> k::provable( W  ->  Z );
> k::provable( W0 & Z  ->  W );
> k::provable( W{p0 & box(box p0 -> p0)/p0} & box box W  ->  Grz1 );
> k::provable( box A4 & Z  ->  Dum1 );
```

Miscellaneous relations:

```
> k::provable( R{box p0 -> p1/p0}  ->  Lp );
> k::provable( T{dia box p0/p0} & R  ->  Zem );
> k::provable(   box box T{A4/p0}
               & box box T{~(box dia box p0 & box p0)/p0}
               & box A4
               & Lp{dia box p0/p0, ~ box p0/p1} & Zem
               ->  R );
> k::provable(   T{dia p0/p0} & box T{~ box dia p0/p0}
               & A4{dia p0/p0}
               & box P  ->  M );
> k::provable( box box T{~p0/p0} & P  ->  R );
> k::provable( A4{~ box p0/p0} & box M & R  ->  P );
> k::provable( box box box T & P & box P{~p0/p0}  ->  Hs );
> k::provable(   T{box(p0 -> box p0) -> p0/p0}
               & box A4{p0 & (box(p0 -> box p0) -> p0)/p0}
               & R{p0 -> box p0/p0}
               ->  Dum );
```

Relations concerning Dum, Grz:

```
> k::provable(    A4{box(p0 -> box p0) -> p0/p0} & box A4
                & Dum & Dum{p0 -> box p0/p0}
                ->  Dum1 );
> k::provable( T & Dum1  ->  Dum );
> k::provable( box T & Dum  ->  Dum2 );
> k::provable( T & box T{p0 -> box p0/p0} & Dum2  ->  Dum );
> k::provable(    box T
                & A4{box(p0 -> box p0) -> p0/p0} & box A4
                & Dum & Dum{p0 -> box p0/p0}
                ->  Dum3 );
> k::provable(   T & box T{p0 -> box p0/p0} & Dum3  ->  Dum);
> k::provable( Dum  ->  Dum4 );
> k::provable( T & Dum4  ->  Dum );
> k::provable( Grz{(box(p0 -> box p0) -> p0) & A4{box(p0 -> box p0) -> p0/p0}/p0}
             & box Grz
             ->  Grz1 );
> k::provable( T & T{Grz1/p0} & box Grz1  ->  Grz );
> k::provable( Grz & box Grz  ->  Grz2 );
> k::provable(   box T{p0 -> box p0/p0}
             & T{~ box(box(p0 -> box p0) -> p0)/p0}
             & Grz2
             ->  Grz );
> k::provable(   Grz{(box(p0 -> box p0) -> box p0) & Grz
                   & A4{(box(p0 -> box p0) -> box p0) & Grz/p0}/p0}
             & box Grz
             ->  Grz3 );
> k::provable( T & box T{p0 -> box p0/p0} & Grz3  ->  Grz );
> k::provable(
       Grz
     & Grz{(box(p0 -> box p0) -> p0) & A4{box(p0 -> box p0) -> p0/p0}/p0}
     ->  Grz4 );
> k::provable( T & Grz4  ->  Grz );
> k::provable(
       Grz{(box(p0 -> box p1) -> box p1)
           & A4{box(p0 -> box p1) -> box p1/p0}/p0}
     & Grz{(box((p0 -> box p1) -> box(p0 -> box p1)) -> (p0 -> box p1))
           & A4{box((p0 -> box p1) -> box(p0 -> box p1)) -> (p0 -> box p1)/p0}/p0}
     & box Grz{ p0 -> box p1/p0}
     ->  Grz5 );
> k::provable(   T{box(p0 -> box p0) -> p0/p0}
             & box box T{p0 -> box p0/p0}
             & A4{box(p0 -> box p0) -> p0/p0} & box A4
             & Grz5{p0 -> box p0/p1}  ->  Grz );
> k::provable( Grz  ->  T );
> k::provable( Grz{p0 & A4/p0}  ->  A4 );
> k::provable(   Grz{dia ~p0/p0}
             & Grz{dia ~p0 & A4{dia ~p0/p0}/p0}
             & Grz{~ box(p0 -> box p0) & A4{~ box(p0 -> box p0)/p0}/p0}
             & box Grz
             ->  M );
> k::provable( Grz  ->  Dum );
> k::provable( Grz1  ->  Dum1 );
> k::provable( box T{~p0/p0} & M & Dum  ->  Grz );
```

11.5 SUMMARY

We have shown that it is possible, with the help of the LWB, to prove the most important relations between normal modal logics. The results can be presented in a uniform way, which makes it easy to check them.

12

COUNTING FORMULAS

'I am great headman, Ghân-buri-Ghân. I count many things: stars in sky, leaves on trees, men in the dark. You have a score of scores counted ten times and five. They have more!'

J.R.R. Tolkien. The Lord of the Rings.

12.1 INTRODUCTION

In all the logics we consider in this thesis, there are infinitely many formulas. Even if we divide the set of formulas into classes of equivalent formulas, we still have an infinite number of equivalence classes.

In this chapter we investigate fragments of the logics. Instead of all formulas we only have formulas that can be built from a set of atoms and a set of connectives. If we take a finite set of variables as atoms, then the number of equivalence classes in CPC is finite. The situation is more complicated in the case of modal logics.

12.2 THEORY

12.2.1 DEFINITION fragment

A fragment F is a triple $\langle L, \mathit{atoms}, \mathit{connectives}\rangle$, where:

- $L \in \{\mathsf{K}, \mathsf{KT}, \mathsf{S4}\}$.
- *atoms* is a finite subset of non-equivalent elements of Fml_L (i.e. $\mathit{atoms} \subseteq \mathrm{Fml}_L$ and $\forall A, B \in \mathit{atoms} : L \not\models A \leftrightarrow B$).
- *connectives* is a set of unary and binary connectives of the logic L.

Example

$\langle \mathsf{K}, \{p, \Diamond q\}, \{\Box, \rightarrow\}\rangle$ is a fragment.

12.2.2 DEFINITION formulas of a fragment

The set Fml_F of formulas of a fragment $F = \langle L, \mathit{atoms}, \mathit{connectives}\rangle$ is defined inductively:

- $A \in \mathit{atoms} \;\Rightarrow\; A \in \mathrm{Fml}_F$
- $\star \in \mathit{connectives}$, $\star$ unary, $A \in \mathrm{Fml}_F \;\Rightarrow\; \star A \in \mathrm{Fml}_F$
- $\circ \in \mathit{connectives}$, $\circ$ binary, $A \in \mathrm{Fml}_F$, $B \in \mathrm{Fml}_F \;\Rightarrow\; A \circ B \in \mathrm{Fml}_F$

Example

If $F = \langle \mathsf{K}, \{p, q\}, \{\Box, \rightarrow\}\rangle$, then:

$$\begin{aligned}\mathrm{Fml}_F \;=\; & \{p, q,\\ & \Box p, \Box q, p \rightarrow p, p \rightarrow q, q \rightarrow p, q \rightarrow q,\\ & \Box\Box p, \ldots, (q \rightarrow p) \rightarrow \Box q, \ldots\}\end{aligned}$$

12.2.3 DEFINITION maximal set of non-equivalent formulas

A set S is a maximal set of non-equivalent formulas of a fragment F if:

- $S \subseteq \mathrm{Fml}_F$
- $A, B \in S \;\Rightarrow\; L \not\models A \leftrightarrow B$
- $A \in \mathrm{Fml}_F \;\Rightarrow\; \exists B \in S : L \models A \leftrightarrow B$

Example

The following two sets are maximal sets of non-equivalent formulas of $\langle \mathsf{S4}, \{p, q\}, \{\wedge, \vee\}\rangle$:

- $\{p, q, p \wedge q, p \vee q\}$
- $\{p, q \wedge q, q \wedge p, q \vee q \vee p\}$

12.2.4 THEOREM size of finite maximal ...

If S_1 and S_2 are finite maximal set of non-equivalent formulas of a fragment, then $\mathrm{card}(S_1) = \mathrm{card}(S_2)$.

Proof

The theorem is a consequence of the 'transitivity' of $\leftrightarrow$, i.e. the fact that $L \models A \leftrightarrow B$ and $L \models B \leftrightarrow C$ imply $L \models A \leftrightarrow C$ for $L \in \{\mathsf{K}, \mathsf{KT}, \mathsf{S4}\}$.

12.2.5 REMARK compute maximal ...

At the moment it is not clear how to compute a maximal set of non-equivalent formulas for a given fragment. Of course we can start with the atoms, and then construct new formulas with the help of the connectives in the fragment, but it is not clear when we can stop, i.e. whether or not the set constructed so far is maximal. Theorem 12.2.8 shows how we can decide this question. We start with two auxiliary definitions.

12.2.6 DEFINITION depth_F

For all fragments $F = \langle L, \mathit{atoms}, \mathit{connectives}\rangle$ we define inductively the function $\mathrm{depth}_F : \mathrm{Fml}_F \rightarrow \mathbb{N}$.

- $A \in \mathit{atoms} \;\Rightarrow\; \mathrm{depth}_F(A) = 0$
- $\star \in \mathit{connectives}$, $\star$ unary, $A \in \mathrm{Fml}_F$, $\star A \notin \mathit{atoms} \;\Rightarrow\; \mathrm{depth}_F(\star A) = 1 + \mathrm{depth}_F(A)$
- $\circ \in \mathit{connectives}$, $\circ$ unary, $A \in \mathrm{Fml}_F$, $B \in \mathrm{Fml}_F$, $A \circ B \notin \mathit{atoms} \;\Rightarrow\; \mathrm{depth}_F(A \circ B) = 1 + \max(\mathrm{depth}_F(A), \mathrm{depth}_F(B))$

Example

If $F = \langle \mathsf{K}, \{p, q, \Box p, \Box q\}, \{\Diamond, \wedge, \vee\}\rangle$, then:

$$\begin{array}{rcl}
\mathrm{depth}_F(q \wedge \Diamond(p \vee \Box q)) & = & 1 + \max(\mathrm{depth}_F(q), \mathrm{depth}_F(\Diamond(p \vee \Box q))) \\
& = & 1 + \max(0, \mathrm{depth}_F(\Diamond(p \vee \Box q))) \\
& = & 1 + \max(0, 1 + \mathrm{depth}_F(p \vee \Box q)) \\
& = & 1 + \max(0, 1 + 1 + \max(\mathrm{depth}_F(p), \mathrm{depth}_F(\Box q))) \\
& = & 1 + \max(0, 1 + 1 + \max(0, 0)) \\
& = & 3
\end{array}$$

12.2.7 DEFINITION $A \in_{L\models} S$

If $L \in \{\mathsf{K}, \mathsf{KT}, \mathsf{S4}\}$, and $S \subseteq \mathrm{Fml}_F$, then:

$$A \in_{L\models} S \quad :\Leftrightarrow \quad \exists B \in S : L \models A \leftrightarrow B$$

12.2.8 THEOREM characterisation of maximal . . .

Let S be a finite set of non-equivalent formulas of a fragment $F = \langle L, \mathit{atoms}, \mathit{connectives}\rangle$.
If

- $\mathit{atoms} \subseteq S$
- $\forall \star \in \mathit{connectives} : \forall A \in S : (\star \text{ unary} \Rightarrow \star A \in_{L\models} S)$
- $\forall \circ \in \mathit{connectives} : \forall A, B \in S : (\circ \text{ binary} \Rightarrow A \circ B \in_{L\models} S)$

then S is maximal.

Proof

Assume that S is not maximal, i.e. that there is a formula C such that $C \notin_{L\models} S$. Then there is a $D \in \mathrm{Fml}_F$ such that $D \notin_{L\models} S$ and $\forall E \in \mathrm{Fml}_F : E \notin_{L\models} S \Rightarrow \mathrm{depth}_F(D) \leq \mathrm{depth}_F(E)$.
D can have three forms:

- $D \in \mathit{atoms}$: Then $D \in S$. Contradiction.
- $D \equiv \star D_1$, $D \notin \mathit{atoms}$: Then $D_1 \notin_{L\models} S$, and $\mathrm{depth}_F(D_1) < \mathrm{depth}_F(D)$. Contradiction.
- $D \equiv D_1 \circ D_2$, $D \notin \mathit{atoms}$: Then $D_1 \notin_{L\models} S$, and $\mathrm{depth}_F(D_1) < \mathrm{depth}_F(D)$. Contradiction.

12.3 ALGORITHM

Theorem 12.2.8 shows a way to compute a maximal set of non-equivalent formulas of a fragment, provided that these sets are finite.

In a first attempt we obtain the following algorithm:

```
$all\_fmls := atoms$
do
    $tmp := \{\star A \mid A \in all\_fmls, \star \in connectives, \star \text{ unary}\}$
            $\cup \{A \circ B \mid A, B \in all\_fmls, \circ \in connectives, \circ \text{ binary}\}$
    $new\_fmls := \emptyset$
    foreach $A \in tmp$ do
        if $A \notin_{L\models} all\_fmls$ and $A \notin_{L\models} new\_fmls$
        then $new\_fmls := new\_fmls \cup \{A\}$
    $all\_fmls := all\_fmls \cup new\_fmls$
until $new\_fmls = \emptyset$
```

In this algorithm *tmp* is too large, since after the first loop it contains formulas we generated already in the previous loop. We improve the algorithm by introducing another set *last_fmls* that contains the formulas that were generated in the previous loop.

```
$all\_fmls := atoms$
$last\_fmls := \emptyset$
do
    $tmp := \{\star A \mid A \in last\_fmls, \star \in connectives, \star \text{ unary}\}$
            $\cup \{A \circ B \mid A \in all\_fmls, B \in last\_fmls, \circ \in connectives, \circ \text{ binary}\}$
            $\cup \{A \circ B \mid A \in last\_fmls, B \in all\_fmls, \circ \in connectives, \circ \text{ binary}\}$
    $new\_fmls := \emptyset$
    foreach $A \in tmp$ do
        if $A \notin_{L\models} all\_fmls$ and $A \notin_{L\models} new\_fmls$
        then $new\_fmls := new\_fmls \cup \{A\}$
    $all\_fmls := all\_fmls \cup new\_fmls$
    $last\_fmls := new\_fmls$
until $new\_fmls = \emptyset$
```

As a further improvement we can make use of the commutativity of binary connectives.

12.4 IMPLEMENTATION

Compared with the algorithm in the previous section 12.3, there are three differences:

- We use lists instead of sets.
- We do not first construct a list `tmp`, but check immediately whether we have to insert it in `new_fmls` for each constructed formula.
- The search for new formulas stops if `limit` formulas have been found. This makes it possible to investigate the 'beginning' of infinite maximal sets of non-equivalent formulas of a fragment.

```
> proc : fmls(atoms, connectives, limit)
   #
    local new_fmls, last_fmls;
    local total, x;
```

```
#
##############################
#
# Check whether  fml  is equivalent to a formula we have constructed before.
# If not, insert  fml  into  new_fmls .
#
 proc insert(fml)
 local x;
 begin
   foreach x in all_fmls do if provable(x <-> fml) then return;
   foreach x in new_fmls do if provable(x <-> fml) then return;
   append(new_fmls, fml);
   inc(total);
   print(total);
   if ( total >= limit )
   then begin
     all_fmls := concat(all_fmls, new_fmls);
     raiseerror("limit");
   end;
 end;
#
##############################
#
# The main procedure. Constructs the formula and checks with  insert  whether
# they are ‘new’.
#
 proc search()
 local l, n, x, y;
 begin
   n := nops(atoms);
   while n > 0 do begin
     print("   while");
     foreach x in last_fmls do begin
       foreach c in connectives do begin
         if (c = NOT) then insert(~x);
         if (c = BOX) then insert(box x);
         if (c = DIA) then insert(dia x);
       end;
       foreach y in all_fmls do begin
         foreach c in connectives do begin
           if (c = AND) then insert(x & y);
           else if (c = OR) then insert(x v y);
           else if (c = IMP) then begin insert(x -> y); insert(y -> x); end;
           else if (c = EQ) then insert(x <-> y);
         end;
       end;
     end;
     n := nops(new_fmls);
     all_fmls := concat(all_fmls, new_fmls);
     last_fmls := new_fmls;
     new_fmls := [];
```

```
      end;
    end;
   #
   ##############################
   #
  begin
    total := 0;
    new_fmls := []; all_fmls := [];
    foreach x in atoms do insert(x);
    all_fmls := new_fmls; last_fmls := new_fmls; new_fmls := [];
    catcherror(search());
    return(all_fmls);
  end;
```

12.5 RUNS

Modalities are formulas that consist of a sequence of $\Box$ and $\Diamond$ followed by p or $\neg p$. We compute the number of modalities in S4.

```
> load(s4);
> timestart; l := sort(fmls([p,~p], [BOX,DIA], 100)); timestop;
```

Thus there are 14 non-equivalent modalities in S4 (cp. for example [Che80]).

In the case of KT, there are infinitely many non-equivalent modalities; in fact no two modalities are equivalent (see [Bel89]). We compute the first 15 non-equivalent modalities.

```
> load(kt);
> timestart; l := sort(fmls([p], [BOX,DIA], 15)); timestop;
```

In K, there are also infinitely many non-equivalent modalities. We compute the first 20 non-equivalent modalities.

```
> load(k);
> timestart; l := sort(fmls([p,~p], [BOX,DIA], 20)); timestop;
```

12.6 OTHER LOGICS

12.6.1 THEOREM $\langle \mathsf{S5}, \{p_0, \ldots, p_n\}, \{\Box, \Diamond, \neg, \wedge, \vee\}\rangle$

The fragment $\langle \mathsf{S5}, \{p_0, \ldots, p_n\}, \{\Box, \Diamond, \neg, \wedge, \vee\}\rangle$ is finite for all n.

12.6.2 THEOREM $\langle \mathsf{CPC}, \{p_0, \ldots, p_n\}, \{\neg, \wedge, \vee\}\rangle$

There are 2^{2^n} equivalence classes in the fragment $\langle \mathsf{CPC}, \{p_0, \ldots, p_n\}, \{\neg, \wedge, \vee\}\rangle$.

12.6.3 THEOREM $\langle \mathsf{CPC}, \{p_0, \ldots, p_n\}, \{\rightarrow\}\rangle$

There are $\sum_{k=1}^{n}((-1)^{k-1}\binom{n}{k}2^{2^{n-k}})$ equivalence classes in the fragment $\langle \mathsf{CPC}, \{p_0, \ldots, p_n\}, \{\rightarrow\}\rangle$.

Proof

See [Dar93].

12.6.4 THEOREM some fragments of IPC

The number of equivalence classes in fragments of IPC has been thoroughly investigated in [Hen96] and [dJHdL91].

Here we simply give an extract of the tables in [Hen96] and [dJHdL91]. The entry 'large' means that the entry is a natural number, but greater than 10^7. We have been able to compute all numbers given below except 518 and 2134 with the LWB using the brute-force approach presented in this chapter.

connectives	$\{p_0\}$	$\{p_0, p_1\}$	$\{p_0, p_1, p_2\}$
$\{\wedge\}$	1	3	7
$\{\vee\}$	1	3	7
$\{\wedge, \vee\}$	1	4	18
$\{\rightarrow\}$	2	14	large
$\{\wedge, \rightarrow\}$	2	18	large
$\{\vee, \rightarrow\}$	2	∞	∞
$\{\wedge, \vee, \rightarrow\}$	2	∞	∞
$\{\neg\}$	3	6	9
$\{\wedge, \neg\}$	5	23	311
$\{\vee, \neg\}$	7	385	large
$\{\wedge, \vee, \neg\}$	7	626	large
$\{\rightarrow, \neg\}$	6	518	large
$\{\wedge, \rightarrow, \neg\}$	6	2134	large
$\{\vee, \rightarrow, \neg\}$	∞	∞	∞
$\{\wedge, \vee, \rightarrow, \neg\}$	∞	∞	∞

12.7 SUMMARY

We have shown how decision procedures for K, KT, S4 can be used to count the number of equivalence classes in fragments of these logics. The same brute-force method will be used in the following two chapters.

13

OPTIMAL SIMPLIFICATION

"I quite agree with you," said the Duchess; "and the moral of that is — 'Be what you seem to be' — or if you'd like it put more simply — 'Never imagine yourself not to be otherwise than what it might appear to others that what you were or might have been was not otherwise than what you had been would have appeared to them to be otherwise.' "

L. Carroll. Alice's Adventures in Wonderland.

13.1 INTRODUCTION

Assume that for a given fragment we want to compute the shortest formula A that satisfies a certain property X.

The simplest algorithm is to enumerate all formulas (of this fragment) with length 1. If one of these formulas satisfies the property X, then we stop. Otherwise we enumerate all formulas with length 2, then those with length 3, and so on, until we find a solution.

Because of the huge number of formulas in most interesting fragments, this procedure is only feasible if a very short solution exists. Consider for example the number of formulas in the fragment $\langle \mathsf{S4}, \{\mathsf{true}, \mathsf{false}, p_0\}, \{\Box, \Diamond, \neg, \wedge, \vee, \rightarrow, \leftrightarrow\}\rangle$ with length n. We define $\#_n := \mathrm{card}(\{A \mid A \in \langle \mathsf{S4}, \{\mathsf{true}, \mathsf{false}, p_0\}, \{\Box, \Diamond, \neg, \wedge, \vee, \rightarrow, \leftrightarrow\}\rangle, \mathrm{length}(A) = n\})$.

These numbers can be recursively defined as follows:

- $\#_1 := \mathrm{card}(\{\mathsf{true}, \mathsf{false}, p_0\})$, i.e. $\#_1 = 3$.
- $\#_2 := \mathrm{card}(\{\Box\mathsf{true}, \Diamond\mathsf{true}, \neg\mathsf{true}, \Box\mathsf{false}, \Diamond\mathsf{false}, \neg\mathsf{false}, \Box p_0, \Diamond p_0, \neg p_0\})$, i.e. $\#_2 = 9$.
- $\#_{n+2} := 3 \cdot \#_{n+1} + \sum_{i=1}^{n}(4 \cdot \#_i \cdot \#_{(n+1)-i})$.

We obtain the following values:

n	$\#_n$
1	3
2	9
3	63
4	405
5	3051
6	23409
7	188487
8	1551069
9	13056147
10	111648537

Of course these numbers are smaller if we consider less connectives and only p_0 as an atom, but the problem remains the same. Moreover, we would no longer find the shortest solution.

Assume that we investigate the fragment $\langle L, V, C\rangle$. If the property X satisfies the condition

$$X(A) \text{ and } L \models A \leftrightarrow B \qquad \Rightarrow \qquad X(B)$$

then there is a much more efficient way to compute a solution: We do not enumerate all the formulas, but only the non-equivalent ones (cp. chapter 12).

The following table shows that we obtain much smaller numbers than above. The numbers in the table are for the fragment $\langle \mathsf{S4}, \{\mathsf{true}, \mathsf{false}, p_0\}, \{\Box, \Diamond, \neg, \wedge, \vee, \rightarrow, \leftrightarrow\}\rangle$, but here only the first few numbers would change if we omitted the atoms true and false and the connectives $\Diamond$, $\vee$, $\rightarrow$, $\leftrightarrow$.

n	
1	3
2	3
3	4
4	8
5	29
6	66
7	84
8	204
9	509
10	898

An important property X that satisfies the condition is

$$X(A) \quad :\Leftrightarrow \quad L \models A \leftrightarrow C$$

for $L \in \{\mathsf{K}, \mathsf{KT}, \mathsf{S4}\}$ and a given constant formula C in Fml_L.

Using this property, we can implement a — relative to length — optimal simplification of a formula A. Of course this is only possible for short formulas with few variables, in contrast to the simplification described in chapter 8.

13.2 THEORY

The underlying theory is almost the same as in chapter 12. We just have to replace the function depth_F by an analogous function length_F.

13.3 ALGORITHM

The basic idea is the same as in chapter 12: We start with the given variables, and then construct larger formulas from the already constructed formulas and the given connectives. Those formulas that are equivalent to a formula constructed earlier are eliminated.

In chapter 12, we use the fact that a formula $A \circ B$ has depth $n > 0$ iff one of A, B has depth $n - 1$. Thus it was sufficient to maintain just two lists of formulas during the construction. Here we have to maintain a list of formulas for each length, since we have $\text{length}_F(A \circ B) = n$ iff $\text{length}_F(A) + \text{length}_F(B) = n - 1$ (if $n > 2$ and $A \circ B$ is not an atom of the fragment). Unary connectives cause no problems: $\text{length}_F(\star A) = n$ iff $\text{length}_F(A) = n - 1$ (if $n > 1$ and $\star A$ is not an atom of the fragment).

13.4 IMPLEMENTATION

We do not go into details since (besides the differences discussed in the previous section) the implementation is essentially the same as in chapter 12. Note that the first argument `aim` is a procedure that has to check whether a formula is a solution. Here `aim` will always check whether its argument is equivalent to a given formula.

```
> proc : shorter_fml(aim, max_length, atoms, connectives)
    #
  local fmls_size, found, i, j, n, s, total;
    #
    ############################################################
    #
    proc insert_and_test(var l, a)
    local a1, fmls_i, i;
    begin
      if found = 1 then return 0;
      for i := 1 to fmls_top do
      begin
        fmls_i := fmls[i];
        for j := 1 to nops(fmls_i) do
          if provable(fmls_i[j] <-> a) then return 0;
      end;
      foreach a1 in l do
        if provable(a1 <-> a) then return 0;
      append(l, a);
      if aim(a)
      then begin
        print("found: ", a);
        found := 1;
      end;
      return 0;
    end; # insert_and_test
    #
    ############################################################
    #
    # Construct formulas with unary main connective in  c  and argument in  l .
    #
    proc unary_combinations(l)
    local a, r;
    begin
      r := [];
      foreach a in l do
        foreach c in connectives do
        begin
        if c = NOT then insert_and_test(r, ~a);
        if c = BOX then insert_and_test(r, box a);
        if c = DIA then insert_and_test(r, dia a);
      end;
      return r;
    end; # unary_combinations
    #
    ############################################################
    #
    # Construct formulas with binary main connective in  c  and arguments
    # in  l1  resp.  l2 .
    #
    proc binary_combinations(l1, l2)
    local a1, a2, r;
    begin
      r := [];
      foreach a1 in l1 do foreach a2 in l2 do
        foreach c in connectives do
        begin
          if c = AND then insert_and_test(r, a1 & a2);
          if c = OR  then insert_and_test(r, a1 v a2);
          if c = IMP then insert_and_test(r, a1 -> a2);
          if c = EQ  then insert_and_test(r, a1 <-> a2);
        end;
      return r;
    end; # binary_combinations
    #
    ############################################################
    #
  begin
  found := 0;
   #
  fmls := array[max_length];
   #
  if ( max_length > 1 )
  then begin
    s := [];
    fmls_top := 0;
    insert_and_test(s, true);
    insert_and_test(s, false);
    for i := 1 to nops(atoms) do insert_and_test(s, atoms[i]);
    fmls[1] := convert2array(s);
```

```
end;
 #
for i := 2 to max_length do
begin
  if ( found = 0 )
  then begin
    fmls_top := i - 1;
    s := unary_combinations(fmls[i - 1]);
    for j := 1 to i - 2 do
      s := concat(s, binary_combinations(fmls[j], fmls[i - (1 + j)]));
    fmls[i] := convert2array(s);
  end;
end;
 #
print();
if ( found = 0 ) then return "nothing found";
 #
end; # shorter_fml
```

13.5 RUNS

For each formula A in figure 1.a, we try to find a shorter formula B such that $\mathsf{S4} \models A \leftrightarrow B$.

For formulas with length less than 8 and one variable, the search is fast (for example less than ten seconds for M). With the LWB 1.0, a complete search is possible for all one-variable formulas from figure 1.a.
We start with some simple formulas that are not valid in S4, namely 5, B, M, and Z.

```
> load(s4);
> A5 :== dia p0 -> box dia p0;
  proc : aim(a) begin return s4::provable(a <-> A5); end;
> shorter_fml(aim, 5, [p0], [BOX,DIA,NOT,AND,OR,IMP,EQ]);
> B :== p0 -> box dia p0;
  proc : aim(a) begin return s4::provable(a <-> B); end;
> shorter_fml(aim, 4, [p0], [BOX,DIA,NOT,AND,OR,IMP,EQ]);
> M :== box dia p0 -> dia box p0;
  proc : aim(a) begin return s4::provable(a <-> M); end;
> timestart;
  shorter_fml(aim, 6, [p0], [BOX,DIA,NOT,AND,OR,IMP,EQ]);
  timestop;
> Z :== box(box p0 -> p0) -> (dia box p0 -> box p0);
  proc : aim(a) begin return s4::provable(a <-> Z); end;
> timestart;
  shorter_fml(aim, 11, [p0], [BOX,DIA,NOT,AND,OR,IMP,EQ]);
  timestop;
```

In figure 13.a we list the results for all one-variable formulas from figure 1.a. If the table entry for a formula A is 'no such formula', then an exhaustive search was carried out with the LWB and no solution was found, i.e. there exists no formula B that is shorter than A and $\mathsf{S4} \models A \leftrightarrow B$.

A short discussion of these results:

formula	shorter equivalent formula in $\mathsf{S4}$
D	true
D_2	true
T	true
4	true
5	no such formula
B	no such formula
G	no such formula
M	M_2
M_2	no such formula
Pt	M_2
W	$\Box p_0$
W_0	false
Z	$\Box p_0 \leftrightarrow \Diamond\Box p_0$
Dum	no such formula
Dum_1	no such formula
Dum_2	$\Diamond\Box p_0 \to \Box(\Box(p_0 \to \Box p_0) \to p_0) \to p_0$
Dum_3	$\Box p_0 \leftrightarrow \Diamond\Box p_0 \wedge \Box(\Box(p_0 \to \Box p_0) \to p_0)$
Dum_4	$\Diamond\Box p_0 \to \Box(\Box(p_0 \to \Box p_0) \to p_0) \to p_0$
Grz	no such formula
Grz_1	no such formula
Grz_2	Grz
Grz_3	$\Box(\Box(p_0 \to \Box p_0) \to p_0) \leftrightarrow \Box p_0$
Grz_4	Grz
H_s	no such formula
P	no such formula
R	no such formula
X	true
Zem	no such formula

Figure 13.a: Results of search for shorter equivalent formula in $\mathsf{S4}$.

- The formulas D, D_2, T, 4 are valid in S4, and thus the corresponding entries in the table must be true. For 5 and B no shorter formulas exist.
- G cannot be shortened.
- We have S4 $\models M \leftrightarrow M_2$ and S4 $\models M \leftrightarrow Pt$ (cp. theorem 11.3.6). Thus the result for M and Pt is M_2. The formula M_2 itself cannot be shortened.
- Obviously W_0 is equivalent to false in S4. Since S4 $\models \Box p_0 \to p_0$ the formula W can be simplified to $\Box p_0$. For the same reason Z can be simplified to $\Diamond\Box p_0 \to \Box p_0$. With the search we find the equivalent formula $\Box p_0 \leftrightarrow \Diamond\Box p_0$.
- No formula is shorter than Dum and equivalent to Dum in S4. Since S4 $\models \mathrm{Dum} \leftrightarrow \mathrm{Dum}_2$ and S4 $\models \mathrm{Dum} \leftrightarrow \mathrm{Dum}_4$ (cp. theorem 11.3.9) the result for Dum_2 and Dum_4 is essentially Dum. Also Dum_1 cannot be simplified, and it is equivalent to Dum_3 and to the formula $\Box p_0 \leftrightarrow \Diamond\Box p_0 \wedge \Box(\Box(p_0 \to \Box p_0) \to p_0)$ found by our search. For Grz, Grz_1, Grz_2, Grz_3, Grz_4 we have the same situation as for Dum, Dum_1, Dum_2, Dum_3, Dum_4. The formula $\Box(\Box(p_0 \to \Box p_0) \to p_0) \leftrightarrow \Box p_0$ found for Grz_3 is equivalent to Grz_1.
- None of the formulas H_s, P, R, Zem can be shortened. X is valid in S4 as it is an instance of the formula T.

13.6 SUMMARY

We have shown how to search for the shortest formula A in $\mathrm{Fml}_{\mathsf{S4}}$ with S4 $\models A \leftrightarrow C$, where C is a given formula. This (relative to length) optimal simplification is only feasible if C is short and contains very few variables. Nevertheless it has, for example, been possible to prove with the LWB that the formulas Grz and Dum cannot be simplified in S4.

14

DIAGRAMS

Heterogeneous logical systems are logics having both linguistic and non-linguistic elements, such as diagrams, charts, tables, etc. They are very important with respect to our main thesis since a great deal of inference involving non-linguistic representations combines the non-linguistic with the linguistic.

J. Barwise and E. Hammer. Diagrams and the Concept of Logical System.

14.1 INTRODUCTION

In chapter 12 we generated maximal sets of non-equivalent formulas. We use the following ordering:

$$A \preceq_L B \;:\Leftrightarrow\; L \models A \to B$$

A set for formulas of Fml_L together with the ordering $\preceq_L$ can be drawn as a graph. In order to obtain simpler pictures, we use the following two conventions:

- We draw just lines, not arrows. The direction is always from bottom to top.
- Obviously the ordering $\preceq_L$ is reflexive and transitive for the logics we consider. Therefore we draw only those lines that do not follow with reflexivity/transitivity from other lines.

Take for example the set $\{p_0 \vee p_1, p_1 \wedge p_0, p_0, p_1\}$ of formulas of CPC. Then we have:

- $p_0 \vee p_1 \preceq_{\mathsf{CPC}} p_0 \vee p_1$
- $p_0 \preceq_{\mathsf{CPC}} p_0,\;\; p_0 \preceq_{\mathsf{CPC}} p_0 \vee p_1$
- $p_1 \preceq_{\mathsf{CPC}} p_1,\;\; p_1 \preceq_{\mathsf{CPC}} p_0 \vee p_1$
- $p_1 \wedge p_0 \preceq_{\mathsf{CPC}} p_1 \wedge p_0,\;\; p_1 \wedge p_0 \preceq_{\mathsf{CPC}} p_0,\;\; p_1 \wedge p_0 \preceq_{\mathsf{CPC}} p_1,\;\; p_1 \wedge p_0 \preceq_{\mathsf{CPC}} p_0 \vee p_1$

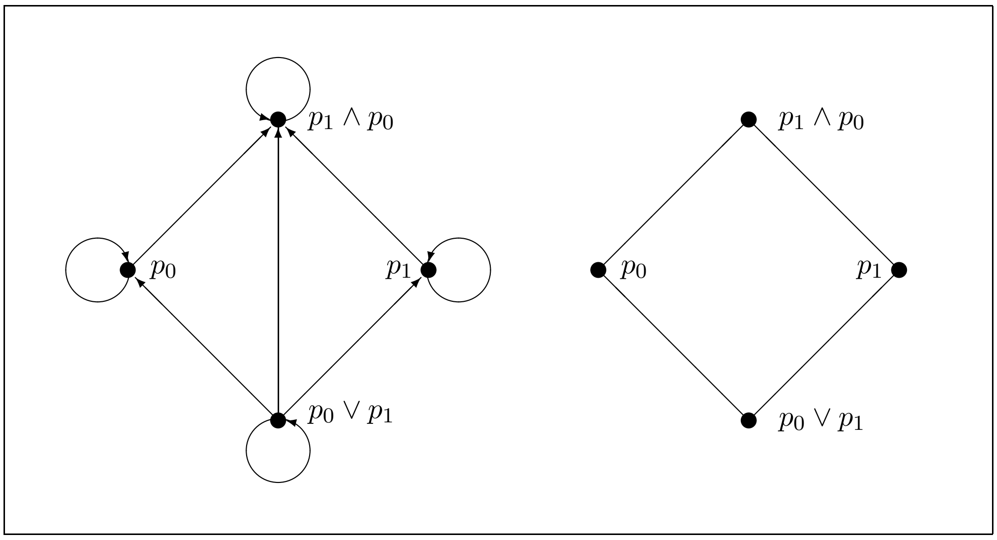


Figure 14.a: The same diagram with and without arrows that correspond to reflexivity and transitivity.

If we drew all arrows, then we would obtain the diagram on the left hand side of figure 14.a. If we drew lines instead of arrows and omitted all lines that follow with reflexivity/transitivity, then we would obtain the one on the right hand side of this figure.

To draw such diagrams automatically is a difficult task. The diagrams in this chapter have been drawn 'by hand'.

14.2 ALGORITHM

In a first step we compute a maximal set of non-equivalent formulas of the given fragment (cp. chapter 12).

Then we compute the diagram by extracting the minimal elements of this list of formulas iteratively. For each extracted formula we compute the list of its predecessors. For this purpose it is sufficient to consider only formulas that have been extracted in a previous step.

```
$old\_min := []$
while $fmls \neq []$
do
    $min := \{A \in fmls \mid \forall B \in fmls, B \neq A : L \models B \to A\}$
    foreach $A \in min$
    do
        $pred := \{B \in old\_min \mid L \models B \to A$
                    and $\forall C \in old\_min, C \neq B : (L \not\models B \to C$ or $L \not\models C \to A)\}$
        print($A$," above ", $pred$)
    $old\_min := min$
    $fmls := \{A \in fmls \mid A \notin min\}$
```

14.3 IMPLEMENTATION

We use the function `fmls` from the last chapter, but without output during the construction of the formulas.

```
> proc : fmls(atoms, connectives, limit)
    local new_list, last_list;
    local total, x;
   #
    proc insert(fml)
    local x;
    begin
      foreach x in all_list do if provable(x <-> fml) then return;
      foreach x in new_list do if provable(x <-> fml) then return;
      append(new_list, fml);
      inc(total);
      if ( total >= limit )
      then begin
        all_list := concat(all_list, new_list);
        raiseerror("limit");
      end;
    end; # insert
   #
    proc search()
    local l, n, x, y;
    begin
      n := nops(atoms);
      while n > 0 do begin
        foreach x in last_list do begin
          foreach c in connectives do begin
            if (c = NOT) then insert(~x);
            if (c = BOX) then insert(box x);
            if (c = DIA) then insert(dia x);
          end;
          foreach y in all_list do begin
            foreach c in connectives do begin
              if (c = AND) then insert(x & y);
              else if (c = OR) then insert(x v y);
              else if (c = IMP) then begin insert(x -> y); insert(y -> x); end;
              else if (c = EQ) then insert(x <-> y);
            end;
          end;
        end;
        n := nops(new_list);
        all_list := concat(all_list, new_list);
        last_list := new_list;
        new_list := [];
      end;
    end; # search
   #
```

```
  begin
    total := 0;
    new_list := []; all_list := [];
    foreach x in atoms do insert(x);
    all_list := new_list; last_list := new_list; new_list := [];
    catcherror(search());
    return(all_list);
  end;
```

In order to obtain a readable output, we introduce natural numbers as abbreviations for the formulas generated by `fmls` .

```
> # add_index  converts a list  [a1, a2, ..., an]  into  [[1,a1], [2,a2], ..., [n,an]] .
   #
  proc : add_index(var fmls)
   #
  local a, i, r;
   #
  begin
    r := [];
    i := 0;
    foreach a in fmls do begin inc(i); append(r, [i,a]); end;
    fmls := r;
  end; # add_index
```

`numbers` is the inverse procedure of `add_index` .

```
> # If  fmls  is  [[i1,a1], [i2,a2], ..., [in,an]], then the result is  [i1, i2, ..., in] .
   #
  proc : numbers(fmls)
   #
  local r, x;
   #
  begin
   #
    r := [];
    foreach x in fmls do append(r, x[1]);
    return r;
   #
  end;
```

The procedure `minimal_elements` removes the minimal elements from `fmls` (a variable parameter) and returns them as a list.

```
> proc : minimal_elements(var fmls)
   #
  local flag, i, j, r, fmls2, fmls_rest;
   #
  begin
   #
    r := [];
    fmls2 := convert2array(fmls); # for faster access
    fmls_rest := [];
    for i := 1 to nops(fmls2)
    do begin
      j := 1;
      flag := true;
      while ( j <= nops(fmls2) ) and flag
      do begin
        if ( i <> j )
        then
          if provable(fmls2[j][2] -> fmls2[i][2]) then flag := false;
        inc(j);
      end;
      # if  flag  is true then  fmls2[i]  is minimal
      if flag then append(r, fmls2[i]); else append(fmls_rest, fmls[i]);
    end;
    fmls := fmls_rest;
    return r;
   #
  end; # minimal_elements
```

The procedure `predecessors` computes the predecessors of the formula `x` in the list `fmls` .

```
> proc : predecessors(x, fmls)
   #
  local flag, r, r2, y, z;
   #
  begin
   #
    r := [];
    foreach y in fmls
    do
      if provable(y[2] -> x[2]) then append(r, y);
   #
    # Now we remove superfluous elements.
    r2 := [];
    foreach y in r
    do begin
      flag := false;
      foreach z in r do
        if y <> z then if provable(y[2] -> z[2]) then flag := true;
      if not flag then append(r2, y);
    end;
   #
    return r2;
   #
  end; # predecessors
```

Finally the main procedure `diagram` .

```
> proc : diagram(fmls)
   #
  local x, min, old_min, level;
   #
  begin
   #
```

```
    add_index(fmls);
   #
    min := minimal_elements(fmls);
    print();
    print("level 0");
    foreach a in min do print("  ", a[1], ": ", a[2], ", []");
    print("------------------------------------------------------------");
   #
    level := 1;
    old_min := min;
   #
    while ( fmls <> [] )
    do begin
      print("level ", level);
      min := minimal_elements(fmls);
      foreach x in min
      do begin
        pred := predecessors(x, old_min);
        print("  ", x[1], ": ", x[2], ", ", numbers(pred));
      end;
      print("------------------------------------------------------------");
      inc(level);
      old_min := concat(old_min, min);
    end;
   #
  end; # diagram
```

14.4 RUNS

First we compute the diagram of the modalities of S4 (cp. chapter 12). Note that this computation takes only about half a second (same hardware as described in chapter 10).

```
> load(s4);
> timestart; l :== sort(fmls([p0,~p0], [BOX,DIA], 100)); diagram(l); timestop;
```

Figure 14.b shows the corresponding digram. It consists of two symmetric, disconnected halves.

Since the number of non-equivalent modalities of KT is infinite (see section 12.5), we cannot compute the complete diagrams. We compute the first 3, 7, 15 modalities of KT ending in p_0 and draw the corresponding diagrams (see figures 14.c and 14.d).

```
> load(kt);
> l :== sort(fmls([p0], [BOX,DIA], 3));
> diagram(l);
```

```
> load(kt);
> l :== sort(fmls([p0], [BOX,DIA], 7));
> diagram(l);
```

```
> load(kt);
> l :== sort(fmls([p0], [BOX,DIA], 15));
> diagram(l);
```

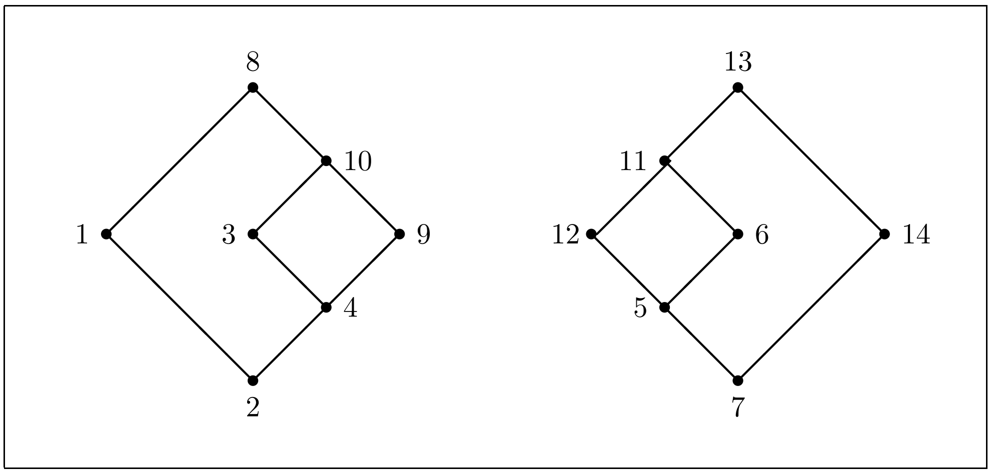


Figure 14.b: The diagram of $\langle \mathsf{S4}, \{p_0, \neg p_0\}, \{\Box, \Diamond\} \rangle$.

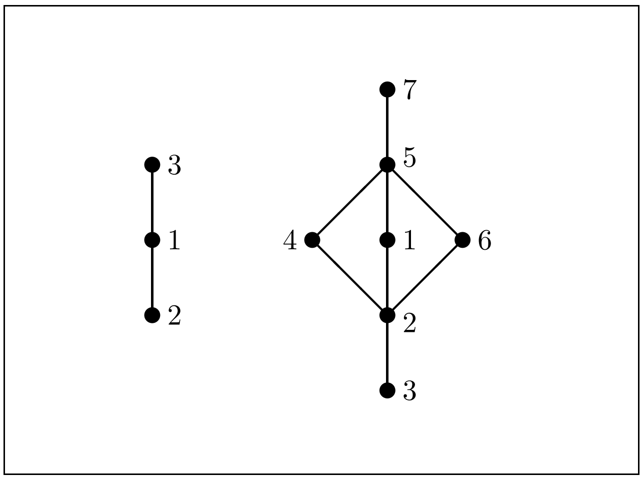


Figure 14.c: The diagrams of the first three and seven modalities of KT.

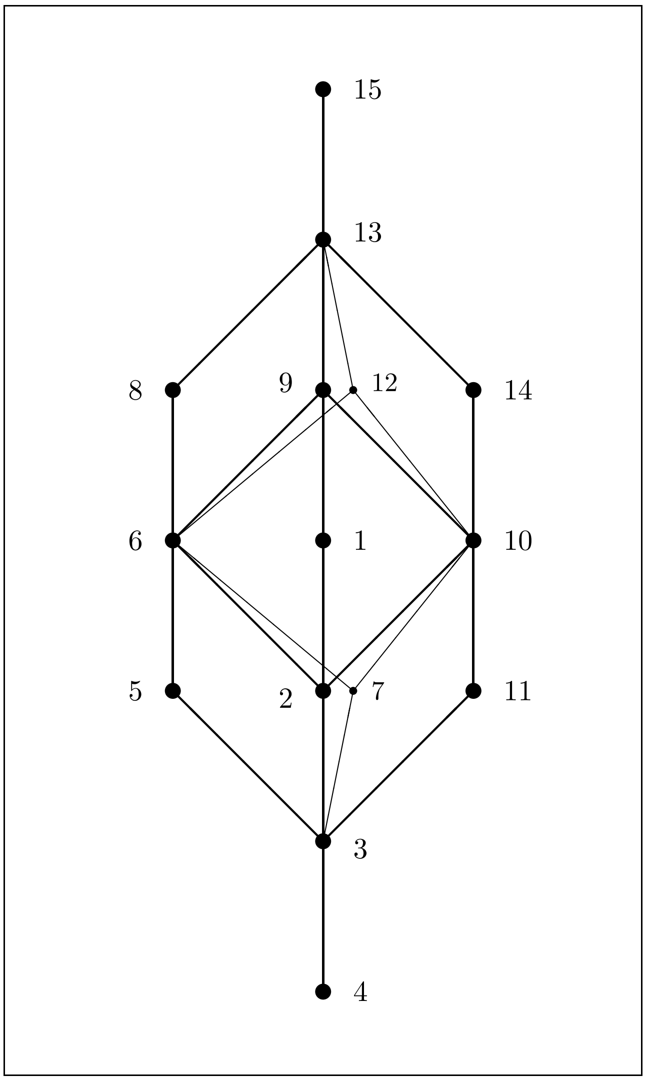


Figure 14.d: The diagram of the first fifteen modalities of KT.

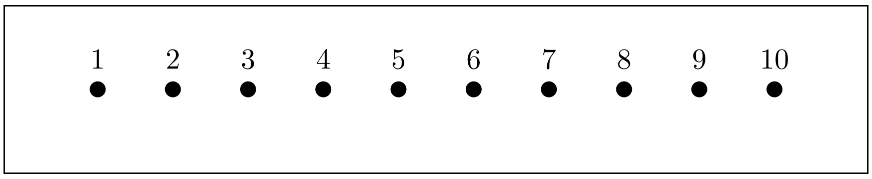


Figure 14.e: The diagram of the first 10 modalities of K.

In K, the diagram of the non-equivalent modalities is trivial, since for all different modalities A and B neither $\mathsf{K} \models A \to B$ nor $\mathsf{K} \models B \to A$ (see figure 14.e).

```
> load(k);
> l :== sort(fmls([p,~p], [BOX,DIA], 10));
> diagram(l);
```

We go back to S4 and compute the diagram of formulas of the form $A \to B$ where A and B are positive modalities. For this purpose we first compute a maximal list of modalities of S4 as before, and then combine them to implications of the requested form.

```
> load(s4);
> l :== sort(fmls([p0], [BOX,DIA], 100));
> l2 :== [];
> foreach x in l do
    foreach y in l do
    begin
      found :== 0;
      foreach z in l2 do if s4::provable(z <-> (x -> y)) then found :== 1;
      if found = 0 then l2 :== concat(l2, [x -> y]);
    end;
> diagram(l2);
```

The resulting diagram looks rather odd (see figure 14.f). Note the numerous hexagons, for example 1-10-27-26-22-21, 1-10-9-26-25-21, 1-11-12-8-9-10, 1-11-7-8-9-10, 1-11-19-20-22-21, and 1-11-12-20-22-21.

Finally we compute the diagram of all positive S4 formulas. Again — although the formulas are far from trivial — only about one minute is required. See figure 14.g for the resulting diagram. We used thick and thin lines in the drawing as if it were a three dimensional structure.

```
> load(s4);
> timestart;
  l :== sort(fmls([p0], [BOX,DIA,AND,OR], 100));
  diagram(l);
  timestop;
```

14.5 OTHER LOGICS

14.5.1 THEOREM diagram of $\langle \mathsf{CPC}, \{p_0, \ldots, p_n\}, \{\wedge, \vee, \neg\} \rangle$

The diagram of $\langle \mathsf{CPC}, \{p_0, \ldots, p_n\}, \{\wedge, \vee, \neg\} \rangle$ is an n-dimensional cube.

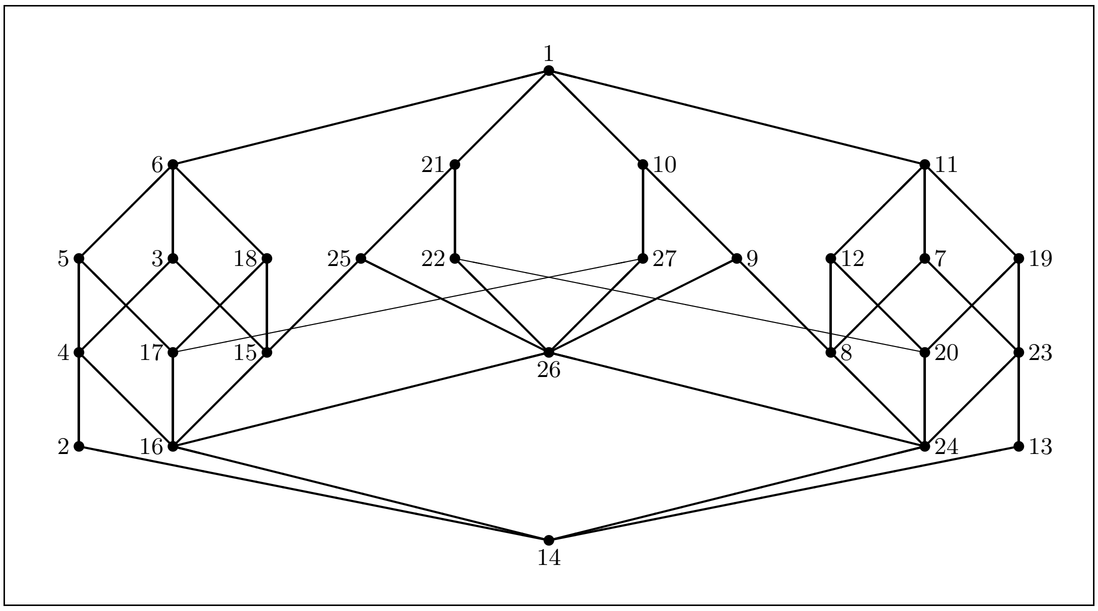


Figure 14.f: The diagram of the implications $A \rightarrow B$, where A, B are positive modalities of $\mathsf{S4}$.

14.5.2 REMARK diagram of $\langle \mathsf{IPC}, \{\mathsf{false}, p_0\}, \{\wedge, \vee, \rightarrow\}\rangle$

The diagram of $\langle \mathsf{IPC}, \{\mathsf{false}, p_0\}, \{\wedge, \vee, \rightarrow\}\rangle$ is the well-known (infinite) Rieger-Nishimura lattice (see for example [TvD88]).

14.5.3 REMARK diagram of fragments of IPC

Diagrams of fragments of IPC are thoroughly discussed in [Hen96].

14.6 SUMMARY

We have used the decision procedures and the programming language of the LWB to compute diagrams of fragments of the modal logics K, KT, S4. In spite of the brute-force approach we have used, all the computations have been possible in a short time.

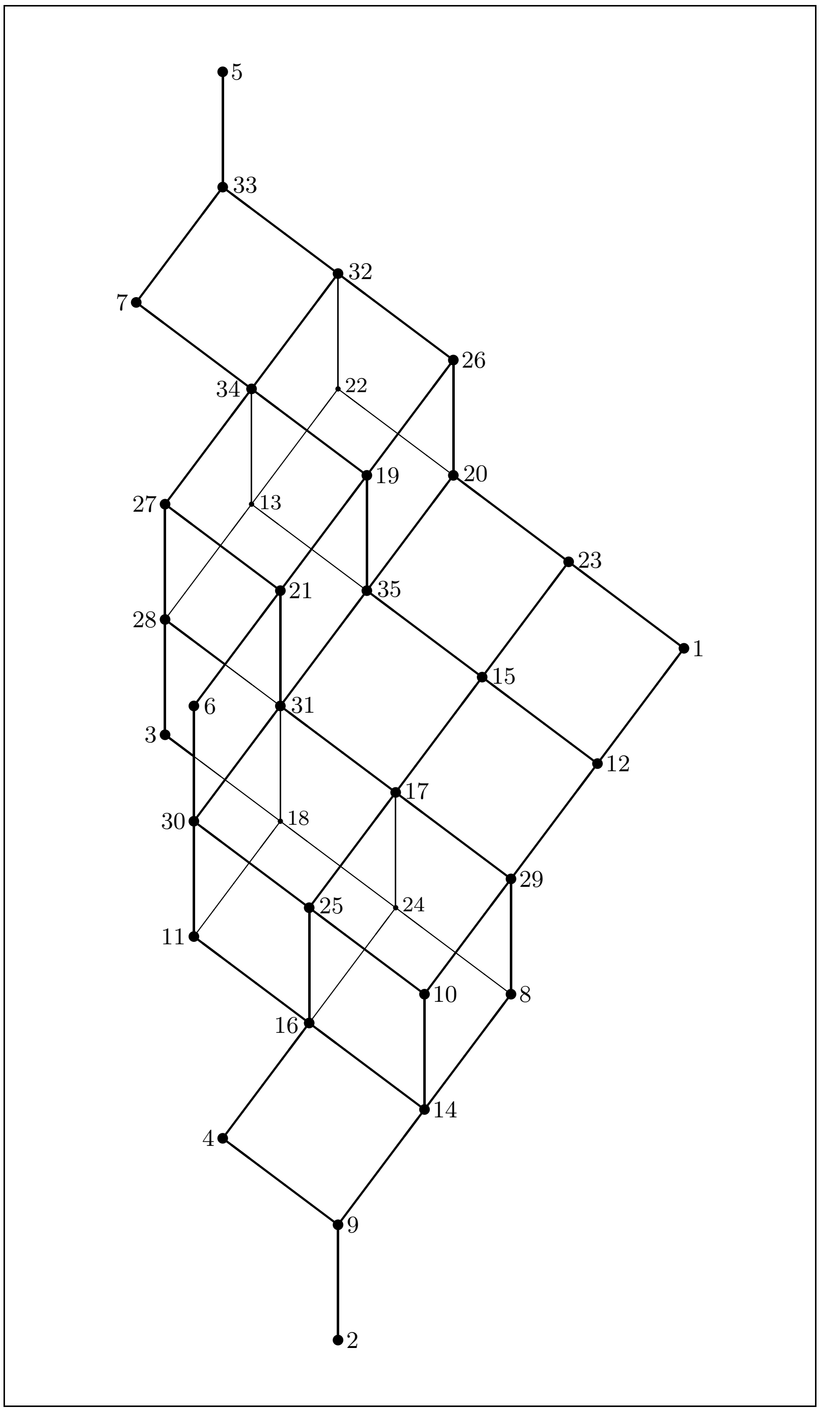


Figure 14.g: The diagram of the positive formulas of S4.

CONCLUSION

'Could you not begin at the beginning and go on until you come to the end, and then, if you are able to, stop?'
'I'll try,' said his lordship, 'but I always find the stopping part of the business so difficult.'

D.L. Sayers. Murder must advertise.

BIBLIOGRAPHY

'Do they read Shakespeare?' asked the Savage as they walked, on their way to the Biochemical Laboratories, past the School Library.
'Certainly not,' said the Head Mistress, blushing.
'Our library,' said Dr Gaffney, 'contains only books of reference.'

A. Huxley. Brave New World.

[And92] JEAN-MARC ANDREOLI.
Logic programming with focusing proofs in linear logic.
Journal of Logic and Computation, 2(3):297–347, 1992.

[BE93] W. BIBEL AND E. EDER.
Methods and calculi for deduction.
In D. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 1, pages 68–182. Clarendon Press, Oxford, 1993.

[Bel89] F. BELLISSIMA.
Infinite sets of nonequivalent modalities.
Notre Dame Journal of Formal Logic, 30(4):574–582, 1989.

[BG97] B. BECKERT AND R. GORÉ.
Free variable tableaux for propositional modal logics.
In D. Galmiche, editor, *Tableaux 97*, LNCS 1227, pages 91–106, 1997.

[BH91] F. BAADER AND B. HOLLUNDER.
A terminological knowledge representation system with complete inference algorithms.
In *PDK 91*, LNAI 567, pages 67–86, 1991.

[BS86] R. BULL AND K. SEGERBERG.
Basic modal logic.
In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic. Volume 2: Extensions of Classical Logic*. Reidel, Dordrecht, 1986.

[BS97] R. Bornat and B. Sufrin.
Jape: A calculator for animating proof-on-paper.
In W. McCune, editor, *CADE-14*, LNAI 1249, pages 412–415, 1997.

[Cat91] L. Catach.
Tableaux: A general theorem prover for modal logics.
Journal of Automated Reasoning, 7:489–510, 1991.

[CG97] S. Clemente and M. Gaspari.
VALE*: evaluating strategies for strongly analytic tableaux.
Position paper at the Tableaux 97 conference, 1997.

[Che80] B. Chellas.
Modal logic: an introduction.
Cambridge University Press, 1980.

[CM97] S. Cerrito and M. Mayer.
Hintikka multiplicities in matrix decision methods for some propositional modal logics.
In D. Galmiche, editor, *Tableaux 97*, LNAI 1227, pages 138–152, 1997.

[CZ97] A. Chagrov and M. Zakharyaschev.
Modal Logic.
Clarendon Press, Oxford, 1997.

[d'A92] M. d'Agostino.
Are tableaux an improvement on truth-tables?
Journal of Logic, Language and Information, 1(3):235–252, 1992.

[Dar93] D. Darms.
Implikation in der mehrwertigen Logik.
Master's thesis, ETH Zürich, Switzerland, 1993.

[Dem95] S. Demri.
Uniform and non uniform strategies for tableaux calculi for modal logics.
Journal of Applied Non-Classical Logics, 5(1):77–96, 1995.

[dG95] P. de Groote.
Linear logic with Isabelle: Pruning the search tree.
In P. Baumgartner, R. Hähnle, and J. Posegga, editors, *Tableaux 95*, LNAI 918, pages 263–277, 1995.

[dJHdL91] D. de Jongh, A. Hendriks, and G. Renardel de Lavalette.
Computations in fragments of intuitionistic propositional logic.
Journal of Automated Reasoning, 7:537–561, 1991.

[Dos93] K. Dosen.
Modal translations in K and D.
In M. de Rijke, editor, *Diamonds and defaults*, pages 103–127. Kluwer, 1993.

[dS98] H. de Swart, editor.
Tableaux 98, LNCS, 1998.

[Dyc91] R. Dyckhoff.
MacLogic. A proof assistant for first-order logic on the Apple Macintosh, 1991.

[Dyc92] R. Dyckhoff.
Contraction-free sequent calculi for intuitionistic logic.
The Journal of Symbolic Logic, 57(3):795–807, 1992.

[Eme90] E. Emerson.
Temporal and modal logic.
In J. v. Leeuwen, editor, *Handbook of Theoretical Computer Science. Volume B*, pages 995–1072. Elsevier, 1990.

[FHMV96] R. Fagin, J. Halpern, Y. Moses, and M. Vardi.
Reasoning about Knowledge.
MIT Press, 1996.

[Fit83] M. Fitting.
Proof Methods for Modal and Intuitionistic Logics.
Reidel, Dordrecht, 1983.

[Fit88] M. Fitting.
First-order modal tableaux.
Journal of Automated Reasoning, 4:191–213, 1988.

[Fit93] M. Fitting.
Basic modal logic.
In D. Gabbay, C. Hogger, and J. Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 1, pages 368–448. Clarendon Press, Oxford, 1993.

[GHH97] R. Goré, W. Heinle, and A. Heuerding.
Relations between propositional normal modal logics: an overview.
Journal of Logic and Computation, 7(5):649–658, 1997.

[Gor] R. Goré.
Tableau methods for modal and temporal logics.
To appear in Handbook of Tableaux Methods.

[Gou89] G. Gough.
Decision procedures for temporal logic.
Technical Report UMCS-89-10-1, Department of Computer Science, University of Manchester, 1989.

[GS96] F. Giunchiglia and R. Sebastiani.
Building decision procedures for modal logics from propositional decision procedures — the case study of modal K.
In M. McRobbie and J. Slaney, editors, *CADE-13*, LNAI 1104, pages 583–597, 1996.

[Häh93] R. Hähnle.
Automated Deduction in Multiple-valued Logics.
Oxford Science Publications, 1993.

[Hal95] J. Halpern.
The effect of bounding the number of primitive propositions and the depth of nesting on the complexity of modal logic.
Artificial Intelligence, 75:361–372, 1995.

[Hei95] W. Heinle.
Expressivity and Definability in Extended Modal Languages.
PhD thesis, TU München, Germany, 1995.

[Hen96] A. Hendriks.
Computations in Propositional Logic.
PhD thesis, Institute for Logic, Language and Information. University of Amsterdam, 1996.

[HM92] J. Halpern and Y. Moses.
A guide to completeness and complexity for modal logics of knowledge and belief.
Artificial Intelligence, 54:319–379, 1992.

[Hor97] I. Horrocks.
Optimising tableaux decision procedures for description logics.
PhD thesis, University of Manchester, 1997.

[How97] J. Howe.
Two loop detection mechanisms: A comparison.
In D. Galmiche, editor, *Tableaux 97*, LNAI 1227, pages 188–200, 1997.

[HS97] U. Hustadt and R. Schmidt.
On evaluating decision procedures for modal logic.
In *IJCAI-97*, 202–207. Morgan Kaufmann, 1997.

[HSZ96] A. Heuerding, M. Seyfried, and H. Zimmermann.
Efficient loop-check for backward proof search in some non-classical propositional logics.
In P. Miglioli, U. Moscato, D. Mundici, and M. Ornaghi, editors, *Tableaux 96*, LNAI 1071, pages 210–225, 1996.

[Hud93] J. Hudelmaier.
An O($n \log n$)-space decision procedure for intuitionistic propositional logic.
Journal of Logic and Computation, 3(1):63–75, 1993.

[Hud96] J. Hudelmaier.
Improved decision procedures for the modal logics K, KT, S4.
In *CSL 95*, LNCS 1092, pages 320–334, 1996.

[Jan90] G. Janssen.
Hardware verification using temporal logic: a practical view.
In L. Claesen, editor, *Formal VSLI Correctness Verification*, pages 159–168. Elsevier, 1990.

[Kas94] R. Kashima.
Cut-free sequent calculi for some tense logics.
Studia Logica, 53:119–135, 1994.

[KP92] E. Koutsoupias and C. Papadimitriou.
On the greedy algorithm for satisfiability.
Information Processing Letters, 43:53–55, 1992.

[Lad77] R. Ladner.
The computational complexity of provability in systems of modal propositional logic.
SIAM journal on computing, 6(3):467–480, 1977.

[Mas94] F. Massacci.
Strongly analytic tableaux for normal modal logics.
In A. Bundy, editor, *CADE 12*, LNCS 814, pages 723–737, 1994.

[Min90] G. Mints.
Gentzen-type systems and resolution. Part I. Propositional logic.
In P. Martin-Löf and G. Mints, editors, *COLOG-88*, LNCS 417, pages 198–231. Springer, 1990.

[MM94] S. Martini and A. Masini.
A modal view of linear logic.
The Journal of Symbolic Logic, 59(3):888–899, 1994.

[MMO95] P. Miglioli, U. Moscato, and M. Ornaghi.
Refutation systems for propositional modal logics.
In P. Baumgartner, R. Hähnle, and J. Posegga, editors, *Tableaux 95*, LNAI 918, pages 95–105, 1995.

[Mor76] C. Morgan.
Methods for automated theorem proving in nonclassical logics.
IEEE Transactions on Computers, C-25(8):852–862, 1976.

[NT87] I. Niemelä and H. Tuominen.
Helsinki logic machine: A system for logical expertise.
Technical report, Digital Systems Laboratory, Department of Computer Science, Helsinki University of Technology, 1987.

[Ohl91] H. Ohlbach.
Semantics based translation methods for modal logics.
Journal of Logic and Computation, 1(5):691–746, 1991.

[OK96] J. Otten and C. Kreitz.
T-string unification: Unifying prefixes in non-classical proof methods.
In P. Miglioli, U. Moscato, D. Mundici, and M. Ornaghi, editors, *Tableaux 96*, LNAI 1071, pages 244–260, 1996.

[OS88] F. Oppacher and E. Suen.
Harp: A tableau-based theorem prover.
Journal of Automated Reasoning, 4:69–100, 1988.

[OS97] H. Ohlbach and R. Schmidt.
Functional translation and second-order frame properties of modal logics.
Journal of Logic and Computation, 7(5):581–603, 1997.

[Ott97] J. Otten.
ileanTAP: An intuitionistic theorem prover.
In D. Galmiche, editor, *Tableaux 97*, LNAI 1227, pages 307–312, 1997.

[Pau94] L. Paulson.
Isabelle: A Generic Theorem Prover.
LNCS 828. Springer, 1994.

[PC96] J. Pitt and J. Cunningham.
Distributed modal theorem proving with KE.
In P. Miglioli, U. Moscato, D. Mundici, and M. Ornaghi, editors, *Tableaux 96*, LNAI 1071, pages 160–176, 1996.

[Pel86] F. Pelletier.
Seventy-five problems for testing automatic theorem provers.
Journal of Automated Reasoning, 2:191–216, 1986.

[Rau79] W. Rautenberg.
Klassische und nichtklassische Aussagenlogik.
Vieweg, 1979.

[RU71] N. Rescher and A. Urquhart.
Temporal Logic.
Springer, 1971.

[SC85] A. Sistla and E. Clarke.
The complexity of propositional linear temporal logic.
Journal of the Association for Computing Machinery, 32(3):733–749, 1985.

[Seg71] K. Segerberg.
An essay in classical modal logic.
Technical report, Uppsala University, 1971.

[SFH92] D. Sahlin, T. Franzén, and S. Haridi.
An intuitionistic predicate logic theorem prover.
Journal of Logic and Computation, 2(5):619–656, 1992.

[SLM92] B. Selman, H. Levesque, and D. Mitchell.
A new method for solving hard satisfiability problems.
In *Proceedings 10th AAAI*, pages 47–59, 1992.

[Sob64] B. Sobociński.
Remarks about axiomatizations of certain modal systems.
Notre Dame Journal of Formal Logic, 5(1):71–80, 1964.

[Sob70] B. Sobociński.
Certain extensions of modal system S4.
Notre Dame Journal of Formal Logic, 11(3):347–368, 1970.

[Spa93] E. Spaan.
The complexity of tense logics.
In M. de Rijke, editor, *Diamonds and defaults*, pages 287–307. Kluwer, 1993.

[SS92] W. Sieg and R. Scheines.
Searching for proofs (in sentential logic).
In L. Burkholder, editor, *Philosophy and the Computer*, pages 137–159. Westview Press, Boulder, San Francisco & Oxford, 1992.

[SSY94] G. Sutcliffe, C. Suttner, and T. Yemenis.
The TPTP problem library.
In A. Bundy, editor, *CADE 12*, LNCS 814, pages 252–266, 1994.

[Sta79] R. Statman.
Intuitionistic logic is polynomial-space complete.
Theoretical Computer Science, 9:67–72, 1979.

[Tam94] T. Tammet.
Proof strategies in linear logic.
Journal of Automated Reasoning, 12:273–304, 1994.

[Tam96] T. Tammet.
A resolution theorem prover for intuitionistic logic.
In M. McRobbie and J. Slaney, editors, *CADE-13*, LNAI 1104, pages 2–16, 1996.

[Tro92] A. Troelstra.
Lectures on Linear Logic.
Number 29 in CSLI Lecture Notes. CSLI, 1992.

[TS96] A. Troelstra and H. Schwichtenberg.
Basic Proof Theory.
Cambridge University Press, 1996.

[TvD88] A. Troelstra and D. v. Dalen.
Contructivism in Mathematics. An Introduction. Volume 1.
North-Holland, 1988.

[vB87] J. v. Benthem.
Transitivity follows from Dummet's axiom.
Theoria, 44:117–118, 1987.

[Vor92] A. Voronkov.
Theorem proving in non-standard logics based on the inverse method.
In *CADE 92*, LNAI 607, pages 648–662, 1992.

[Wal90] L. Wallen.
Automated Proof Search in Non-Classical Logics.
M.I.T. Press, Cambridge, Massachusetts, 1990.

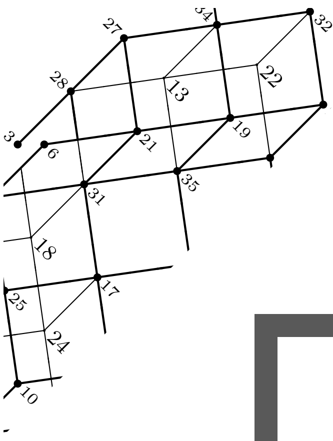
[Wol85] P. Wolper.
The tableau method for temporal logic: an overview.
Logique et Analyse, 110-111:119–136, 1985.

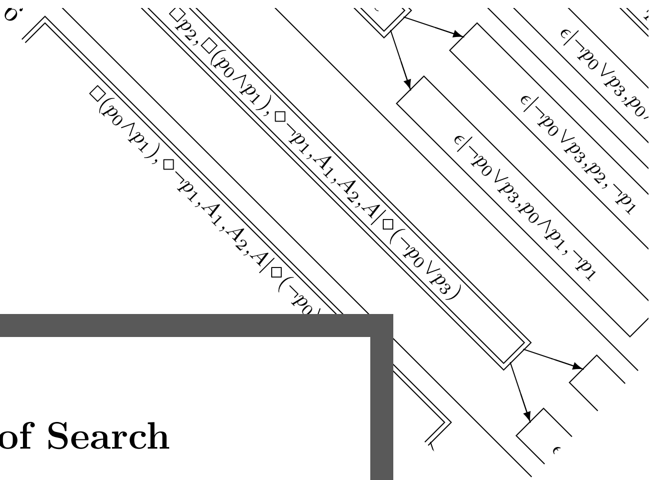
[Zim94] H. Zimmermann.
A directed tree calculus for minimal tense logic.
Master's thesis, IAM, University of Bern, Switzerland, 1994.

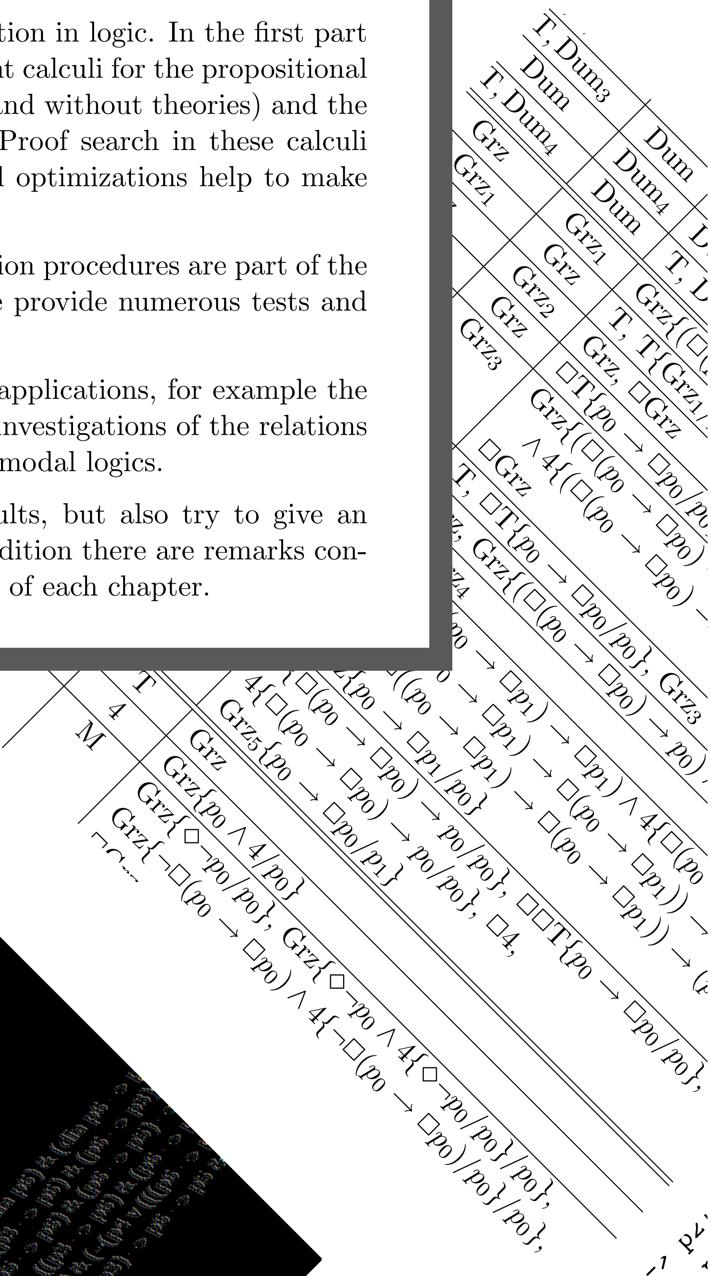
INDEX

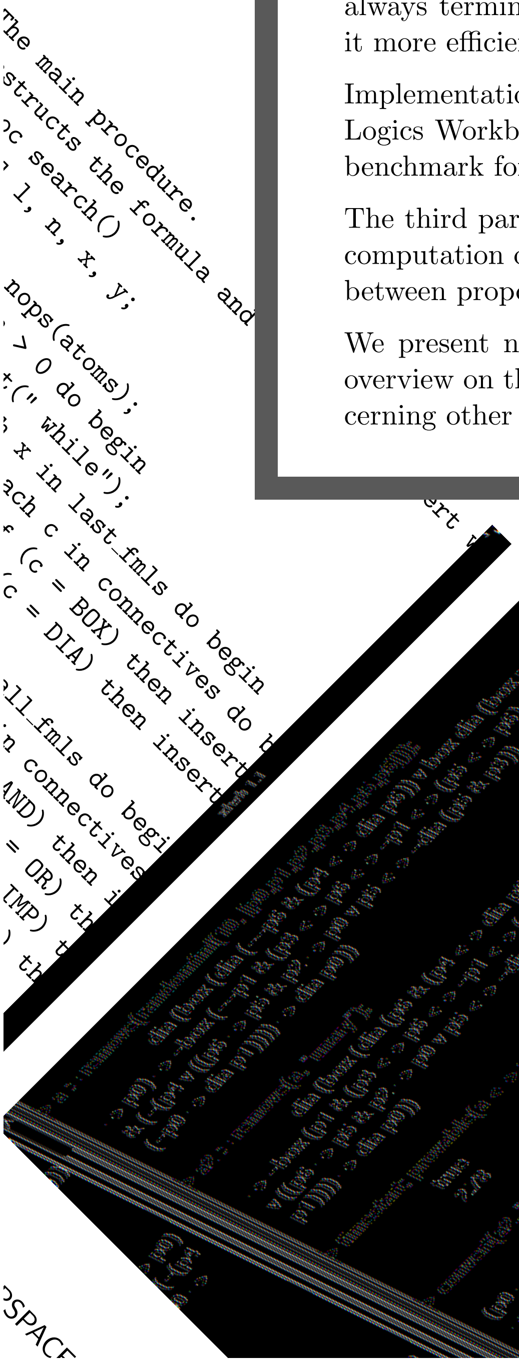
“5 3 ‡ ‡ † 3 0 5)) 6 * ; 4 8 2 6) 4 ‡ .) 4 ‡) ; 8 0 6 * ; 4 8 † 8 § 6 0)) 8 5 ; 1 ‡) ; : ‡ * 8 † 8 3) 8 8) 5 * † ; 4 6) ; 8 8 * 9 6 * ? ; 8) * ‡) ; 4 8 5) ; 5 * † 2 : * ‡) ; 4 9 5 6 * 2) 5 * – 4) 8 § 8 * ; 4 0 6 9 2 8 5) ;) 6 † 8) 4 ‡ ‡ ; 1) ‡ 9 ; 4 8 0 8 1 ; 8 : 8 ‡ 1 ; 4 8 † 8 5 ; 4) 4 8 5 † 5 2 8 8 0 6 * 8 1) ‡ 9 ; 4 8 ;) 8 8 ; 4) ‡ ? 3 4 ; 4 8) 4 ‡ ; 1 6 1 ; : 1 8 8 ; ‡ ? ;”
‘But,’ said I, returning him the slip. ‘I am as much in the dark as ever. Were all the jewels of Golconda awaiting me on my solution of this enigma, I am quite sure that I should be unable to earn them.’
‘And yet,’ said Legrand, ‘the solution is by no means so difficult as you might be led to imagine from the first hasty inspection of the characters.’

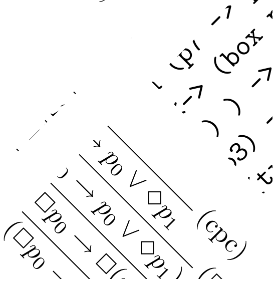
E.A. Poe. The gold-bug.











Sequent Calculi for Proof Search in Some Modal Logics

Modal logics have a long tradition in logic. In the first part of this thesis we develop sequent calculi for the propositional modal logics K, KT, $\mathsf{S4}$ (with and without theories) and the propositional tense logic K_t. Proof search in these calculi always terminates, and several optimizations help to make it more efficient.

Implementations of these decision procedures are part of the Logics Workbench (LWB). We provide numerous tests and benchmark formulas.

The third part contains some applications, for example the computation of diagrams and investigations of the relations between propositional normal modal logics.

We present not only new results, but also try to give an overview on the subject. In addition there are remarks concerning other logics at the end of each chapter.