

Priorean tense logics in modal pure type systems

Citation for published version (APA):

Borghuis, V. A. J. (1996). *Priorean tense logics in modal pure type systems*. (Computing science reports; Vol. 9613). Technische Universiteit Eindhoven.

Document status and date:

Published: 01/01/1996

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Eindhoven University of Technology
Department of Mathematics and Computing Science

Priorean Tense Logics in Modal Pure Type Systems

by

Tijn Borghuis

96/13

ISSN 0926-4515

All rights reserved

editors: prof.dr. R.C. Backhouse
prof.dr. J.C.M. Baeten

Reports are available at:
<http://www.win.tue.nl/win/cs>

Computing Science Report 96/13
Eindhoven, June 1996

Contents

| | | |
|----------|-------------------------------------------------------------------------------------------------|-----------|
| 1 | Introduction | 1 |
| 2 | Tense logic | 3 |
| 2.1 | The basic system K_t | 3 |
| 2.2 | Further properties of time | 4 |
| 2.2.1 | Pure axioms | 4 |
| 2.2.2 | Mixed axioms | 5 |
| 2.3 | Linear flows of time | 6 |
| 2.3.1 | The logic Lin | 6 |
| 2.3.2 | Familiar linear structures | 6 |
| 3 | Natural deduction for tense logics | 8 |
| 3.1 | Fitch-style deduction for multi-modal systems | 8 |
| 3.2 | Natural deduction for K_t | 17 |
| 3.3 | Further properties of the tenses | 18 |
| 3.4 | The interaction between past and future | 21 |
| 3.4.1 | The mirror image property | 21 |
| 3.4.2 | General interaction patterns | 24 |
| 3.5 | Linear tense logics | 28 |
| 3.5.1 | The logic Lin | 28 |
| 3.5.2 | Familiar linear structures | 31 |
| 4 | Tense MPTSs | 33 |
| 4.1 | MPTSs with multiple modalities | 33 |
| 4.2 | The MPTS λK_t | 37 |
| 4.3 | Further properties of the tenses | 39 |
| 4.4 | The interaction between past and future | 41 |
| 4.5 | Linear MPTSs | 44 |
| 4.5.1 | The MPTS λLin | 44 |
| 4.5.2 | Familiar linear structures | 45 |
| 4.6 | Subject reduction | 46 |
| 4.6.1 | Reduction in K_t | 46 |
| 4.6.2 | Reduction in Lin | 48 |
| 5 | Axioms and rules | 51 |
| 6 | Concluding remarks | 54 |
| | Acknowledgements | 55 |
| | Appendix: Interderivability of $\diamond A$ and $\neg \Box \neg A$ | 58 |

Priorean Tense Logics in Modal Pure Type Systems*

Tijn Borghuis
Eindhoven University of Technology[†]

1. Introduction

The aim of this paper is to extend typed λ -calculus with temporal reasoning. Typed λ -calculi have a number of features which make them very suitable for applications to knowledge representation (see [Ahn et al. 1994]), but they lack the possibility for reasoning about situations that change in time which is vital to some of these applications. Simply put, the reason for this inability is that the language of type theory (like that of other major logical formalisms) is atemporal; it was designed to deal with the truth and falsity of propositions *sub specie aeternitatis*. This is in strong contrast with natural language, where temporal properties of states of affairs (such as their duration or temporal order) can be expressed directly through changes in verb-form or ‘tense’: ‘John sings’ is true when John sings now, whereas ‘John sang’ is true when John was singing somewhere in the past.

One of the first attempts to reconcile the discrepancy between the atemporal language of logic and tensed natural language with modern logical means were Arthur N. Prior’s ‘tense logics’ ([Prior 1957], [Prior 1967]). In Priorean Tense Logic, standard propositional logic is extended with four operators that transform an untensed proposition (φ) into a tensed expression:

$G\varphi$: ‘always in the future it is Going to be the case that φ ’
 $F\varphi$: ‘somewhere in the Future it will be the case that φ ’
 $H\varphi$: ‘always in the past it Has been the case that φ ’
 $P\varphi$: ‘somewhere in the Past it was the case that φ ’.

The extended language allows the expression of a great variety of temporal structures and temporal arguments. This has led to widespread applications ranging from the semantics of natural language (see [Gamut 1982]) to the verification of computer programs (cf. [Goldblatt 1992]).

For the purposes of this paper, the main advantage of Priorean Tense Logics is that they capture temporal reasoning by means of intensional (or modal) operators. From the point of view of present day modal logic, tense logics can be conceived of as multi-modal logics of which the operators interact in specific ways. This view provides a direct connection with the framework of Modal Pure Type Systems (MPTSs) presented in [Borghuis 1994]. These systems extend typed λ -calculi that correspond to standard non-modal logics with modal

*This research was carried out in the project ‘Dialogue Handling and Knowledge Acquisition’ (DenK), supported by the Tilburg-Eindhoven Organisation for Inter-University Cooperation

[†]Department of Mathematics and Computing Science, P.O. Box 513, 5600 MB Eindhoven, The Netherlands.
E-mail: tijn@win.tue.nl

operators. For a family of multi-modal logics a ‘propositions-as-types’ interpretation has already been given in the MPTS-framework, i.e. a formal mapping has been constructed from natural deduction proofs in these logics to terms in an MPTS (and vice versa). Therefore investigating the possibilities for the interpretation of Priorean Tense Logics in MPTSs seems a promising way of bringing temporal reasoning into type theory.

Because of the unusual combination of subjects, keeping this paper self-contained is infeasible. However, we do not aim at an (imaginary) ideal reader with detailed knowledge of both Priorean Tense Logic and Modal Pure Type Systems but assume that the reader is familiar with basic modal logic (e.g. [Chellas 1980]) and linear natural deduction (as in [van Westrhenen et al. 1993]) on the one hand and with typed λ -calculi (preferably Pure Type Systems [Barendregt 1992]) and the propositions-as-types interpretation of (non-modal) logic on the other hand. Throughout the paper we try to focus on ideas rather than technical details, and to provide the reader with sufficient examples to gain an intuitive understanding of what is going on.

The structure of this paper is as follows: first we give an introduction to the basic tense logic and a number of its extensions, each capturing a different conception of the flow of time (section 2). Then natural deduction formulations of these logics are given (section 3). Using the natural deduction systems, MPTSs are defined in which the tense logics can be interpreted (section 4). Section 5 contains a short digression on the limitations of strengthening tense logical deduction systems and MPTSs by so-called modal rules. The paper closes with concluding remarks (section 6).

Due to this structure, the reader will meet all of the logical systems and principles discussed in this paper three times: first in an axiomatic guise (section 2), then in a deductive guise (section 3), and finally in the guise of a type system (section 4). Since we treat a fair number of systems and principles, we thought it might be helpful (if only for future reference) to provide the reader with some sort of road-map indicating where particular systems occur and reoccur in the course of this paper. The table below lists (the section numbers of) the three occurrences for each of the main systems and clusters of principles treated in this paper. The first entry is for the minimal Priorean Tense Logic K_t . All other tense logics are obtained by strengthening this logic with additional principles formalizing further properties of time. A number of these principles, listed here as ‘further properties’, are treated in this paper. One of the logics extending K_t is the logic Lin which is the minimal tense logic for linear flows of time. If Lin is in turn extended, we obtain tense logics for a number of linear structures which are familiar from mathematics (like $(\mathbb{N}, <)$). These logics are treated as a group in this paper and listed below as ‘familiar structures’.

| Tense logic | <i>Axioms</i> | <i>Deduction</i> | <i>Type System</i> |
|---------------------|---------------|------------------|--------------------|
| K_t | 2.1 | 3.1 | 4.2 |
| further properties | 2.2 | 3.3 | 4.3 |
| Lin | 2.3.1 | 3.5.1 | 4.5.1 |
| familiar structures | 2.3.2 | 3.5.2 | 4.5.2 |

2. Tense logic

In this section we briefly introduce the tense logical systems that will be subjected to deductive and type theoretical treatment throughout this paper. For a thorough discussion of the listed axioms and logics the reader is referred to [van Benthem 1983].

2.1. The basic system K_t

The language of Priorean Tense Logics is that of (classical) propositional logic extended with operators G , F , H , and P : given a propositional language consisting of propositions letters $A_1, A_2, \dots, B, \dots$, constants \top and \perp , and connectives $\neg, \wedge, \vee, \supset, \leftrightarrow$, $G\varphi$, the formulas $F\varphi$, $H\varphi$ and $P\varphi$ are well-formed if φ is itself a well-formed formula.

The smallest Priorean tense logic is K_t which, according to [Prior 1967], was first proposed by Lemmon in 1965. It can be characterized as the set of propositions derivable by means of:

- all propositional tautologies
- definitions:

$$F\varphi \leftrightarrow \neg G\neg\varphi$$

$$P\varphi \leftrightarrow \neg H\neg\varphi$$

- axioms:

$$G(\varphi \supset \psi) \supset (G\varphi \supset G\psi) \quad H(\varphi \supset \psi) \supset (H\varphi \supset H\psi) \quad (\text{Normality})$$

$$\varphi \supset GP\varphi \quad \varphi \supset HF\varphi \quad (\text{Symmetry})$$

- rules:

$$(\text{Modus Ponens}) \quad \frac{\varphi \quad \varphi \supset \psi}{\psi}$$

$$(\text{Necessitation}_G) \quad \text{if } \varphi \text{ is a thesis, then } G\varphi \text{ is a thesis}$$

$$(\text{Necessitation}_H) \quad \text{if } \varphi \text{ is a thesis, then } H\varphi \text{ is a thesis}$$

(where a *thesis* is a well-formed formula that is an axiom or a theorem of the logic, hence derivable without assumptions; note that *Modus Ponens* also holds in the presence of assumptions).

From the perspective of modal logic, K_t can be seen as consisting of two copies of the minimal normal modal logic K : one for the operator ‘ G ’, looking ‘forward’ in time, and one for the operator ‘ H ’, looking ‘back’. These two logics are linked by the symmetry axioms which ensure that the two directions in which one can look are each other’s inverse: if φ holds now, then going to any point in the future (past) one can see the point where φ holds somewhere in the past (future). In the Kripke models for K_t , the accessibility relation of the operator G , R_G , and that of H , R_H are each the converse of the other: $sR_Gt \Leftrightarrow tR_Hs$ (see for instance [Goldblatt 1992]). Hence the following theorem in [van Benthem 1983] comes as no surprise:

THEOREM. K_t axiomatizes the tense logic of all symmetrical point structures.

An interesting consequence of this symmetry is that K_t has the ‘mirror image property’: if a formula φ in which G and H may occur ($\varphi(G, H)$) is a theorem of K_t , then the same formula with G and H exchanged ($\varphi(H, G)$) is also a theorem of K_t .

2.2. Further properties of time

Tense logics above K_t are obtained by adding axioms expressing further properties of the flow of time. These axioms can be divided into ‘pure’ axioms, which state properties for one direction of time (generally occurring in pairs : one for the future, one for the past), and ‘mixed’ axioms which describe ways in which the past and future direction of time interact.

2.2.1. Pure axioms

Several axioms which are familiar from general introductions to modal logic, such as [Chellas 1980] or [Hughes and Cresswell 1972], reappear when one tries to formalize intuitions about the flow of time. Although pure axioms can be adapted separately for the past and future direction of time, we shall discuss them in ‘mirror-image pairs’ which express a certain intuition symmetrically with respect to past and future.

- $4_f \quad G\varphi \supset GG\varphi \qquad 4_p \quad H\varphi \supset HH\varphi$

These axioms express transitivity of the time flow, which may become clear by looking at the sometimes preferred (e.g. in [Koymans 1989]) K_t -equivalent forms $FF\varphi \supset F\varphi / PP\varphi \supset P\varphi$: in this form 4_f says that any point in the future of a future point is also a future point.

- $D_f \quad G\varphi \supset F\varphi \qquad D_p \quad H\varphi \supset P\varphi$

The operators G and H quantify universally over respectively all future and past time points, regardless of the existence of such point. To ensure that if $G\varphi$ ($H\varphi$) holds there actually exists a future (past) time point at which φ holds, the above axioms for ‘seriality’ or ‘succession’ of the time-point ordering are needed. D_f and D_p enforce that the time flow has no end points in the future and past direction. These principles are expressed in several ways in the literature, such as FT/PT ([Goldblatt 1992]), $GF(\neg\perp)/HP(\neg\perp)$ ([Gabbay et al. 1994]) and $\neg G(\varphi \wedge \neg\varphi)/\neg H(\varphi \wedge \neg\varphi)$ ([Gamut 1982]), all of which are K_t -equivalent to D_f/D_p .

- $T_f \quad G\varphi \supset \varphi \qquad T_p \quad H\varphi \supset \varphi$

These axioms express reflexivity of the time-point ordering: if always in the past (always in the future) φ holds, then φ holds now. Hence adding a T -axiom to K_t turns the ‘<’-ordering of the time points into a ‘ \leq ’-ordering*.

- $Dens_f \quad GG\varphi \supset G\varphi \qquad Dens_p \quad HH\varphi \supset H\varphi$

These axioms express density of the time flow: between any two points in the order there lies another point. This may be more apparent from their K_t -equivalent forms $F\varphi \supset FF\varphi / P\varphi \supset PP\varphi$ ([Koymans 1989]): in this form $Dens_f$ says that any reachable point in the future ($F\varphi$) is also reachable via an intermediate point in the future ($FF\varphi$).

*Clearly we cannot add these axioms separately for the future and past direction. This issue is discussed in section 3.4.

Besides these ‘general purpose’ principles which are also known from applications of modal logic to other areas than temporal reasoning, more subtle intuitions concerning the ordering of the time points can be formalized by means of just G or H .

- $W_f \quad G(G\varphi \supset \varphi) \supset G\varphi \qquad W_p \quad H(H\varphi \supset \varphi) \supset H\varphi$

These tense logical versions of the Löb-axiom from provability logic enforce well-foundedness of the ordering ([van Benthem 1983]): in a given direction there are only finite chains of ordered time points. Hence time flows where time has a beginning can be captured using W_p .

- $Z_f \quad G(G\varphi \supset \varphi) \supset (FG\varphi \supset G\varphi) \qquad Z_p \quad H(H\varphi \supset \varphi) \supset (PH\varphi \supset H\varphi)$

These axioms, also known as ‘modified Löb’ ([van Benthem 1983]), express ‘discreteness’ of the ordering: between any two time points (in a given direction) lie only finitely many points.

- $Dum_f \quad G(G(\varphi \supset G\varphi) \supset \varphi) \supset (FG\varphi \supset \varphi)$
 $Dum_p \quad H(H(\varphi \supset H\varphi) \supset \varphi) \supset (PH\varphi \supset \varphi)$

Proposed by Dummett in 1958, these axioms capture a different idea of discreteness, that of ‘finite variability’: between any two points (in a given direction) a proposition can only go through finitely many changes of its truth value.

2.2.2. Mixed axioms

In the logic K_t , the past and future direction of time are already closely intertwined due to the symmetry axioms. This interaction can be strengthened by adding further axioms ‘mixing’ ‘ H ’ and ‘ P ’ with ‘ G ’ and ‘ F ’.

- $Pres_f \quad F\varphi \supset HF\varphi \qquad Pres_p \quad P\varphi \supset GP\varphi$

These axioms are called ‘preservation axioms’ because they stipulate that existential tense formulas holding for one direction of time are preserved universally in the other direction of time: If $(Pres_p) \varphi$ holds somewhere in the past ($P\varphi$), then in all future points it will be true that φ holds somewhere in the past ($GP\varphi$).

- $L_f \quad F\varphi \supset G(P\varphi \vee \varphi \vee F\varphi) \qquad L_p \quad P\varphi \supset H(P\varphi \vee \varphi \vee F\varphi)$

These axioms enforce linearity of the time flow ‘to the left’ (L_p) and ‘to the right’ (L_f). For instance, L_f preempts branching of the future by demanding that if φ holds at some point in the future ($F\varphi$), any point in the future can ‘see’ this point either ‘in front’ of itself (in its future), ‘behind’ itself (in its past), or in itself. In a branching future this would only hold for the future time points that lie on the same branch as the point at which φ is true.

- $Cont \quad \blacksquare(H\varphi \supset FH\varphi) \supset (H\varphi \supset G\varphi)$

(where ‘ \blacksquare ’ is the modality ‘at any time (past, present or future)’, which is definable in G and H : $\blacksquare\varphi =_{\text{def}} H\varphi \wedge \varphi \wedge G\varphi$.) This axiom, also known as the Inkspot Principle, encodes a Dedekind-like definition of continuity (see [van Benthem 1983], p. 162). It allows us to distinguish between dense and continuous flows of time.

2.3. Linear flows of time

The purpose of this section is to show how K_t can be systematically extended with the above axioms to obtain logical characterizations of different conceptions of the flows of time. We concentrate on linear time flows, which have traditionally been one of the main stays of Priorean analysis. After introducing the basic tense logic underlying all linear time flows, we look at a number of standard examples of linear flows in the literature.

2.3.1. The logic Lin

The logic Lin of all linear flows of time is K_t extended with the axioms for

$$\begin{array}{ll} \text{transitivity: } & 4_f \ G\varphi \supset GG\varphi \qquad 4_p \ H\varphi \supset HH\varphi \\ \text{linearity: } & L_f \ F\varphi \supset G(P\varphi \vee \varphi \vee F\varphi) \\ & L_p \ P\varphi \supset H(P\varphi \vee \varphi \vee F\varphi) \end{array}$$

By adding transitivity to K_t the tense logical theory of strict partial orders is obtained, [van Benthem 1983]. The linearity axioms preclude ‘branching’: they forbid that the past or the future consist of more than one partial order. Hence the combined effect of the axioms is to enforce that all time points (past, present and future) lie on a single ‘time line’.

Throughout the rest of this paper we will use a less customary formulation of the linearity axioms, which can be found in [Goldblatt 1992]:

$$L_f \ \blacksquare\varphi \supset HG\varphi \qquad L_p \ \blacksquare\varphi \supset GH\varphi ,$$

Given the definition of ‘ \blacksquare ’, $\blacksquare\varphi =_{\text{def}} H\varphi \wedge \varphi \wedge G\varphi$, the logical equivalence of the two formulations is provable in K_t . The reason for preferring the ‘ \blacksquare ’-form of the axioms is that they can be given a direct and intuitive treatment in Fitch-style deduction, which is not possible for the axioms in their original formulation (this will be discussed in section 3.5.1).

2.3.2. Familiar linear structures

By extending Lin with different (combinations of) axioms, various conceptions of a linear flow of time can be formalized. In the literature, this is usually illustrated by giving tense-logical characterizations of mathematical number structures, allowing the reader to keep a model in mind in which time points are numbers ordered by the ‘ $<$ ’-relation. We follow this custom and compare tense logics for discrete, dense and real linear time flows.

Integer time $(\mathbb{Z}, <)$

The tense logical theory of integers, $Th_t(\mathbb{Z})$, is axiomatized by Lin plus:

$$\begin{array}{ll} D_f \ G\varphi \supset F\varphi & D_p \ H\varphi \supset P\varphi \\ Z_f \ G(G\varphi \supset \varphi) \supset (FG\varphi \supset G\varphi) & Z_p \ H(H\varphi \supset \varphi) \supset (PH\varphi \supset H\varphi) \end{array}$$

This logic, called $Lin\ Disc$ in [Goldblatt 1992], enforces that time is infinite in both directions by means of D_f and D_p , and that the ordering of time points is discrete by Z_f and Z_p .

Natural time $(\mathbb{N}, <)$

The tense logical theory of the natural numbers, $Th_f(\mathbb{N})$, is axiomatized by Lin plus:

$$\begin{aligned}
D_f & G\varphi \supset F\varphi \\
Z_f & G(G\varphi \supset \varphi) \supset (FG\varphi \supset G\varphi) \\
W_p & H(H\varphi \supset \varphi) \supset H\varphi
\end{aligned}$$

Compared to $Th_t(\mathbb{Z})$, the axioms for the past direction of time have been changed: the axiom D_p expressing infinity in the direction of the past has been replaced by W_p which enforces well-foundedness of the past. Hence in this logic, $Lin Disc^w$ in [Goldblatt 1992], time has a beginning. Note that the ordering of time points in the past direction remains discrete; W_p (propositionally) implies Z_p .

Rational time ($\mathbb{Q}, <$)

The tense logical theory of the rational numbers, $Th_t(\mathbb{Q})$, is axiomatized by Lin plus:

$$\begin{array}{ll}
D_f & G\varphi \supset F\varphi & D_p & H\varphi \supset P\varphi \\
Dens_f & GG\varphi \supset G\varphi & Dens_p & HH\varphi \supset H\varphi
\end{array}$$

This logic, $Lin Rat$, is obtained from that for integer time by replacing the axioms for discreteness (Z_p, Z_f) with those for density of the time-point ordering ($Dens_f, Dens_p$). An even simpler logic can be given for dense *reflexive* time: since both the seriality axioms (D_f, D_p) and the density axioms are derivable given the reflexivity axioms, this structure can be characterized by the logic $Lin Rat Ref$ which is Lin plus:

$$T_f \quad G\varphi \supset \varphi \qquad T_p \quad H\varphi \supset \varphi .$$

Real time ($\mathbb{R}, <$)

$Th_t(\mathbb{R})$ is axiomatized by Lin plus:

$$\begin{array}{ll}
D_f & G\varphi \supset F\varphi & D_p & H\varphi \supset P\varphi \\
Dens_f & GG\varphi \supset G\varphi & Dens_p & HH\varphi \supset H\varphi \\
cont & \blacksquare(H\varphi \supset FH\varphi) \supset (H\varphi \supset G\varphi)
\end{array}$$

In other words, the logic $Lin Re$ for the real numbers is the logic for the rational numbers extended with the ‘Inkspot Principle’ (the axiom *cont*).

3. Natural deduction for tense logics

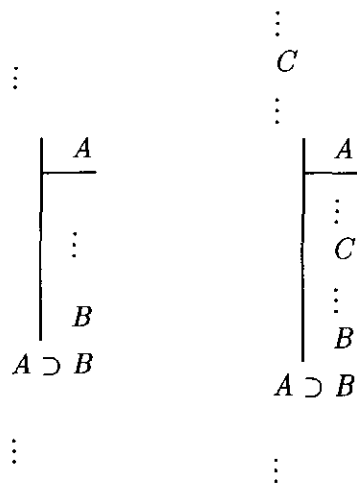
A prerequisite for the type theoretical interpretation of the Priorean tense logics discussed so far is to find a natural deduction formulation for these logics. Although tense logics hardly ever appear in a deductive guise in the literature, we can hope to obtain suitable formulations by using techniques developed for modal logic in general.

In [Borghuis 1994] a framework for multi-modal ‘Fitch-style’ natural deduction is developed in which the tense logics can be stated: we take them to be bi-modal logics of which the operators (G, H) are related by symmetry. After presenting the framework, a natural deduction system for K_t is defined which is then extended in a modular way to deal with stronger logics. Special attention is paid to the effects of the symmetry between the operators in the resulting deduction systems.

3.1. Fitch-style deduction for multi-modal systems

Natural deduction systems for proposition and predicate logic come in two ‘styles’, characterized by the form of their proofs: ‘Prawitz-style’ systems have deduction proofs in the form of trees, ‘Fitch-style’ systems have linear proofs. For modal logic the vast majority of systems in the literature is linear. Fitch-style deduction for modal logic starts in [Fitch 1952], where a new construct is introduced that extends his deduction system for propositional logic to one for modal logic.

Central to Fitch-style propositional deduction is a construction known as ‘subordinate proof’. It consists in writing a proof as part of another proof. For instance, to prove $A \supset B$ one starts a new, subordinate, proof by assuming A and then sets out to prove B . When this goal is achieved the subordinate proof is ended by adding $A \supset B$ to the original proof, justified by the implication introduction rule, thereby discharging the assumption A .



A subordinate proof Reiteration

Structurally (in the graphical representation), subordinate proofs are positioned to the right of the proof to which they are subordinate, the ‘main’ proof. The topmost formula (A) is the *hypothesis* of the subordinate proof, the vertical line indicates the exact extent of the subordinate proof; the *hypothesis interval*.

Subordinate proofs are just like ‘main’ proofs except that some of the formulas in them

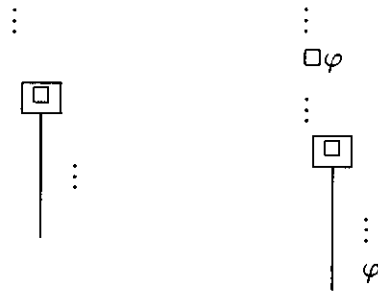
may be repetitions of formulas from a proof to which they are subordinate (in the figure above, C is such a formula). Such a repetition is called ‘reiteration’; a formula in a proof may be reiterated in another proof if the latter is subordinate to the former. Subordinate proofs can be nested at will: a subordinate proof may be written as part of a subordinate proof.

To extend his deduction system to modal logic, Fitch added a new kind of subordinate proof, the *strict* subordinate proof. It differs from ‘ordinary’ subordinate proofs in two respects:

- A strict subordinate proof may be started at any point in a proof, it requires no hypothesis.
- Reiteration in a strict subordinate proof is restricted to formulas of a certain form.

Structurally these proofs are just like subordinate proofs, their ‘strictness’ is indicated by means of a ‘ \square ’ on top of the vertical line, which indicates the *modal interval*.

For the logic K , the logic underlying all normal modal operators, reiteration is restricted to formulas of the general form $\square\varphi$: formulas of this form occurring in a proof may be repeated in a strict subordinate proof, without their boxes (as φ). This procedure can be added to a Fitch-style deduction system for propositional logic in the form of the following rule:

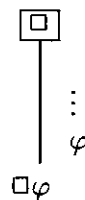


A strict subordinate proof K -import

K-import: φ may occur in a strict subordinate proof if $\square\varphi$ occurs earlier in the proof to which it is immediately subordinate.

A formula that has been imported into a strict subordinate proof never counts as hypothesis of that proof. Strict subordinate proofs may be written as part of another proof, hence we can have arbitrary nestings of strict and ordinary subordinate proofs.

Formulas can also ‘travel’ in the opposite direction: conclusions (φ) derived by means of a *categorical* strict subordinate proof may be added to the main proof in a necessitated form ($\square\varphi$). A subordinate proof is categorical when all its assumptions have been discharged; the conclusion lies directly inside the modal interval, there are no nested subordinate proofs that are still ‘open’. This procedure for ‘exporting’ information from the strict subordinate proof to the main proof is expressed in the following rule:



K-export: if φ occurs in a categorical strict subordinate proof then $\Box\varphi$ may occur later in the proof to which it is immediately subordinate.

Fitch-style systems for multi-modal logics have separate subordinate proofs and K-rules for each of the normal modal operators. Hence for Priorean tense logic we have strict G -subordinate proofs as well as strict H -subordinate proofs. From the temporal perspective, the procedures for import and export can be understood in the following way: if we take a main proof to be the time point at which we try to establish the truth of a tense logical formula, a strict G -subordinate proof (H -subordinate proof) corresponds to an arbitrary future time point (past time point). In such a future time point we only know the truth of the propositions (φ) that hold always in the future ($G\varphi$) of the point of evaluation. In this view, starting a strict G -subordinate proof amounts to continuing the proof in an arbitrary future time point. Every proposition (ψ) that can be derived without hypotheses in such a point could have been derived in any future time point, hence it can be considered to hold always in the future of the original time point ($G\psi$). In this way conclusions obtained in the future time point can be brought back (exported) to the point where the proof was started, and the proof can be resumed there.

We now give a formal definition of $\Box PROP_{fitch}^{\mathcal{O}}$, a Fitch-style deduction system for the multi-modal logic K , by combining a standard Fitch-style system for propositional logic with K -import and K -export*. The system will be presented in the manner of [van Westrhenen et al. 1993], describing the proof figures and deduction rules in terms of intervals. Although the definition is somewhat elaborate, it is more concise than the usual ‘look at the picture’-type of presentation. The benefits of this will become apparent later on, when the vocabulary introduced here allows us to easily describe extensions of the system and to define various notions needed in meta theoretical proofs.

The first stage in defining $\Box PROP_{fitch}^{\mathcal{O}}$ is to specify what configurations of modal and hypothesis intervals are allowed in the Fitch-style modal deduction proofs, given the set of $PROP$ of well-formed formulas of K . Intervals are represented as $[i, j]$, where i and j are the line numbers of the lines in the proof figure that form the extremes of the interval.

3.1. DEFINITION. Proof figure

A *proof figure* \mathbf{D} is a mathematical structure consisting of:

1. a closed interval $D = [1, n]$, where $D \subset \mathbb{N}$,
2. a function $F : D \rightarrow PROP$, and
3. a collection \mathbf{I} of subintervals of D , such that for each interval $[i, j] \in \mathbf{I}$, $i \leq j$, and such that for each pair of (different) intervals $[i, j], [k, l] \in \mathbf{I}$ we have $i < k < l \leq j$, or $k < i < j \leq l$ or $[i, j] \cap [k, l] = \emptyset$. The collection \mathbf{I} of subintervals is the union of two disjoint subcollections \mathbf{H} and \mathbf{M} :

\mathbf{H} the *hypothesis intervals* of the proof figure. If $D \notin \mathbf{H}$, then D is called the *0-th interval*. If $[k, l] \in \mathbf{H}$ then the formula F_k is called *the hypothesis* of $[k, l]$.

\mathbf{M} the *modal intervals* of the proof figure, this set is the union of all sets \mathbf{M}^o where $o \in \mathcal{O}$, the set of operator indices ($\mathcal{O} = \{1, \dots, n\}$ for some $n \in \mathbb{N}$). D may not be an element of \mathbf{M} .

*In [Borghuis 1994] this system is referred to $\Box PROP_{fitch}^{Modalities}$, because there the set \mathcal{O} was called *Modalities*.

In Fitch-style deduction for non-modal propositional logic $\mathbf{I} = \mathbf{H}$; every subinterval is a hypothesis interval introduced by assuming the topmost formula of that interval. The presence of a modal interval in a proof figure does not require an assumption and hence the topmost formula of such an interval is not a hypothesis. Another difference is that a modal interval may never be the leftmost ('0-th') interval of a proof figure: the figure only qualifies as a derivation after all modal subordinate proofs have been closed. In a proof figure a modal interval can be recognized by the box (' \square^o ') on top of its vertical line, where the index o indicates for which of the operators in \mathcal{O} this modal interval is a strict subordinate proof.

Some more terminology is needed before we can define the deduction rules:

3.2. DEFINITION. **Precede, lie in**

If $i \in D$, then $F(i)$, usually written as F_i , denotes the formula on line i of the proof figure. We say that F_i *precedes* F_j , if $i < j$.

If $i \in I$ for a certain interval $I \in \mathbf{I} \cup \{D\}$ and there is no $J \in \mathbf{I}$ such that $i \in J \subset I$, then it is said that the formula F_i *lies in* I , written as $F_i \in I$. An interval I *lies in* interval $J \in \mathbf{I} \cup \{D\}$ if $I \subset J$ and there is no $K \in \mathbf{I}$, such that $I \subset K \subset J$.

To each formula in a proof we attribute a degree of 'nestedness'. In a non-modal system the degree of a formula F_i is simply the number of hypotheses at that stage of the proof: 'the number of vertical lines to the left of the formula' at line i in the proof figure. In modal deduction proofs this set of hypotheses can be 'partitioned' by modal intervals, and for the formal definition of the K -rules we have to keep track of this. Therefore the degree of a formula in a modal proof figure is represented as a *pair* of natural numbers, where the first number denotes the 'modal depth' of F_i : number of nested modal intervals ($\in \mathbf{M}$) 'to the left' of F_i . The second number represents the number of hypothesis intervals ($\in \mathbf{H}$) 'to the right' of the most deeply nested modal interval of which F_i is an element.

3.3. DEFINITION. **Degree**

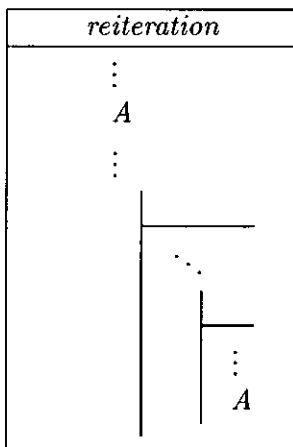
The *degree* of a formula F_i , written $gr(i)$, is defined as a pair of natural numbers:
 $gr(i) = (card\{I \in \mathbf{M} \mid i \in I\}, card\{I \in \mathbf{H}' \mid i \in I\})$ where
 $\mathbf{H}' = \{I \in \mathbf{H} \mid i \in I \text{ and there is no } J \in \mathbf{M} \text{ such that } (i \in J \subset I)\}$
(*card* denotes the cardinality of a set).

The natural deduction rules of $\square PROP_{fitch}^{\mathcal{O}}$ are defined in two stages; first their structural effect on the proof figures is shown in a picture, then the conditions for their application are defined in terms of the form of the formulas acting as the premisses and conclusion of the rule and in terms of the relation between the intervals in which these formulas lie.

3.4. DEFINITION. **Deduction rules**

| | |
|----------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $\begin{array}{c} \text{V-intro} \\ A \qquad B \\ \vdots \qquad \vdots \\ A \vee B \qquad A \vee B \end{array}$ | $\begin{array}{c} \text{V-elim} \\ A \vee B \\ \vdots \\ \begin{array}{ l} A \\ \vdots \\ C \end{array} \\ \vdots \\ \begin{array}{ l} B \\ \vdots \\ C \end{array} \\ \vdots \\ C \end{array}$ |
| $\begin{array}{c} \neg\text{-intro} \\ \begin{array}{ l} A \\ \vdots \\ B \\ \vdots \\ \neg B \end{array} \\ \neg A \end{array}$ | $\begin{array}{c} \neg\text{-elim} \\ \neg\neg A \\ \vdots \\ A \end{array}$ |
| $\begin{array}{c} \supset\text{-intro} \\ \begin{array}{ l} A \\ \vdots \\ B \end{array} \\ A \supset B \end{array}$ | $\begin{array}{c} \supset\text{-elim} \\ \vdots \\ A \supset B \\ \vdots \\ A \\ \vdots \\ B \end{array}$ |
| $\begin{array}{c} \wedge\text{-intro} \\ A \\ \vdots \\ B \\ \vdots \\ A \wedge B \end{array}$ | $\begin{array}{c} \wedge\text{-elim} \\ A \wedge B \qquad A \wedge B \\ \vdots \qquad \vdots \\ A \qquad B \end{array}$ |

| | | |
|---------------------------------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------|
| \leftrightarrow -intro | \leftrightarrow -elim | |
| $A \supset B$ \vdots $B \supset A$ \vdots $A \leftrightarrow B$ | $A \leftrightarrow B$ \vdots $A \supset B$ | $A \leftrightarrow B$ \vdots $B \supset A$ |
| K import | K export | |
| \vdots $\Box^\circ A$ \vdots \Box° \vdots A | \Box° \vdots A \vdots $\Box^\circ A$ | |



In these structural representations of the deduction rules the bottommost formula is the conclusion of the rule, which may be written in a proof if all the premisses indicated above it are already present in the proof. Note that these premisses can be formulas as well as proofs, for instance: the \vee -elim rule has as its premisses one formula ($A \vee B$) and two proofs of C , one under the hypothesis A and one under the hypothesis B .

3.5. DEFINITION. Application of deduction rules

Given a proof figure \mathbf{D} , with interval $D = [1, n]$, formulas F_1, \dots, F_n and intervals \mathbf{I} . A formula E is the result of an *application* of deduction rule R , if E is the conclusion of R , the premisses of R precede E , and one of the following conditions is met:

1. $R \in \{\vee$ -intro, \neg -elim, \supset -elim, \wedge -intro, \wedge -elim, \leftrightarrow -intro, \leftrightarrow -elim $\}$.
 In this case the premisses and the conclusion E all lie in the *same interval*. The order in which the premisses appear may differ from the one given in the table.
2. $R = \neg$ -intro.
 There has to be a hypothesis-interval $[k, l] \in \mathbf{H}$, such that $F_k = A$, and such that either $F_l = \neg B$ and B lies in $[k, l]$, or $F_l = B$ and $\neg B$ lies in $[k, l]$. The conclusion $E = \neg A$

and the interval $[k, l]$ have to lie in the same interval (it is allowed that $B = F_k$ (A and B coincide), or that $\neg B = F_k$ (A and $\neg B$ coincide)).

3. $R = \supset$ -*intro*.

There has to be a hypothesis-interval $[k, l] \in \mathbf{H}$, such that $F_k = A$ and $F_l = B$. The conclusion $E = A \supset B$ and the interval $[k, l]$ have to lie in the same interval.

4. $R = \vee$ -*elim*.

There have to be hypothesis-intervals $[i, j], [k, l] \in \mathbf{H}$, such that $F_i = A$, $F_j = C$, $F_k = B$ and $F_l = C$, where $j < k$, or $l < i$. The conclusion $E = C$, the premiss $A \vee B$ and the intervals $[i, j]$ and $[k, l]$ have to lie in the same interval.

5. $R =$ *reiteration*.

If the premiss A lies in the interval $I \in \mathbf{IU}\{D\}$ and the conclusion $E = A$ lies in the interval $J \in \mathbf{IU}\{D\}$, then it has to be the case that $(J \subseteq I) \wedge \neg \exists K \in \mathbf{M} (J \subset K \subseteq I)$. Or, in terms of modal depth: the first coordinate of $gr(A)$ is equal to the first coordinate of $gr(E)$, and the second coordinate of $gr(A)$ is smaller than or equal to the second coordinate of $gr(E)$.

6. $R = K$ *import*.

If the premiss $\Box^o A$ lies in interval $I \in \mathbf{I}$ where $o \in \mathcal{O}$, and the conclusion $E = A$ lies in the interval $J \in \mathbf{M}^o$, then it has to be the case that *the interval J lies in the interval I* .

7. $R = K$ *export*.

If the premiss A lies in interval $I \in \mathbf{M}^o$ where $o \in \mathcal{O}$, and the conclusion $E = \Box^o A$ lies in the interval $J \in \mathbf{I}$, then it has to be the case that *the interval I lies in the interval J* .

Note that K *export* allows us to export more than one formula from a strict subordinate proof, as long as these formulas all occur after the assumptions in the strict subordinate proof are discharged.

3.6. DEFINITION. **Derivation without hypotheses**

A *derivation* of a formula C is a proof figure \mathbf{D} with interval $D = [1, n]$ and formulas F_1, \dots, F_n , that satisfies the following conditions:

1. $F_n = C$ and $gr(n) = (0, 0)$;
2. every formula $F_i (1 \leq i \leq n)$ is a hypothesis or the result of the application of a deduction rule on a number of formulas preceding F_i .

3.7. DEFINITION. **Derivation with hypotheses**

A *derivation* of a formula C from the formulas P_1, \dots, P_m ($m \geq 1$) is a proof figure \mathbf{D} with interval $D = [1, n]$ ($n > m$) and formulas F_1, \dots, F_n , that satisfies the following conditions:

1. $F_i = P_i$ is a hypothesis for $1 \leq i \leq m$, such that $gr(i) = (0, i)$;
2. $F_n = C$, and C and P_m lie in the same hypothesis-interval, where $gr(n) = (0, m)$
3. every formula $F_i (1 \leq i \leq n)$ is a hypothesis or the result of the application of a deduction rule on a number of formulas preceding F_i .

A derivation with hypotheses is a proof where the assumptions P_1, \dots, P_m are not discharged. These assumptions are listed consecutively at the first m lines of the proof figure, this mandatory enumeration excludes the possibility that there are modal intervals mixed in with the hypothesis intervals.

3.8. DEFINITION. Derivability

1. A formula C is *derivable* if there exists a derivation of C , written as $\vdash C$.
2. A formula C is *derivable* from the formulas P_1, \dots, P_m if there exists a derivation of C from $P_1 \dots P_m$, written as $P_1, \dots, P_m \vdash C$.
3. Let $\Gamma \subseteq PROP$ be a set of formulas. A formula C is *derivable* from Γ if there exist a finite number of formulas $P_1, \dots, P_m \in \Gamma$ such that $P_1, \dots, P_m \vdash C$. This is written: $\Gamma \vdash C$. If $\Gamma = \emptyset$, $\vdash C$.

The multi-modal deduction system we have just finished defining is minimal in the sense that it gives us the smallest normal modal logic K for each of the operators in \mathcal{O} . Nothing is said about further properties of individual operators or interactions between the operators. As will be explained in section 3.3, there are two ways to extend the deduction system to accommodate such strengthenings, one of which is to add modal rules. In [Borghuis 1994] rules are listed for a number of standard mono-modal axioms, which occur throughout literature under various interpretations of the modal operator (see [Chellas 1980]):

$$D : \Box\varphi \supset \neg\Box\neg\varphi$$

$$T : \Box\varphi \supset \varphi$$

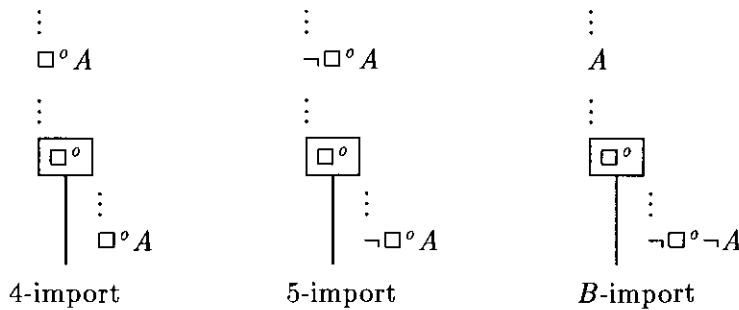
$$4 : \Box\varphi \supset \Box\Box\varphi$$

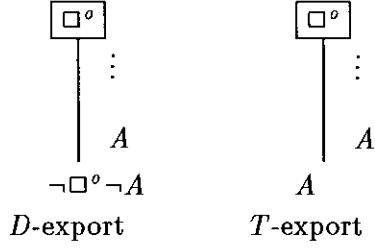
$$5 : \neg\Box\varphi \supset \Box\neg\Box\varphi$$

$$B : \varphi \supset \Box\neg\Box\neg\varphi$$

The reader will have noted that we have already encountered most of these axioms with temporal operators ‘ G ’ and ‘ H ’ replacing ‘ \Box ’, which makes it useful to add the corresponding rules to $\Box PROP_{fitch}^{\mathcal{O}}$. For each of the axioms a single extra import- or export-rule is needed to make it a theorem of the deduction system:

3.9. DEFINITION. Deduction rules





3.10. DEFINITION. Application of deduction rules

Given a proof figure \mathbf{D} , with interval $D = [1, n]$, formulas F_1, \dots, F_n and intervals \mathbf{I} . A formula E is the result of an *application* of deduction rule R , if E is the conclusion of R , the premisses of R precede E , and one of the following conditions is met for the modal rules:

8. $R = 4$ import.
If the premiss $\square^\circ A$ lies in interval $I \in \mathbf{I}$ where $o \in \mathcal{O}_{4 \text{ import}}(\subseteq \mathcal{O})$, and the conclusion $E = \square^\circ A$ lies in the interval $J \in \mathbf{M}^\circ$, then it has to be the case that *the interval J lies in the interval I .*
9. $R = 5$ import.
If the premiss $\neg \square^\circ A$ lies in interval $I \in \mathbf{I}$ where $o \in \mathcal{O}_{5 \text{ import}}(\subseteq \mathcal{O})$, and the conclusion $E = \neg \square^\circ A$ lies in the interval $J \in \mathbf{M}^\circ$, then it has to be the case that *the interval J lies in the interval I .*
10. $R = B$ import.
If the premiss A lies in interval $I \in \mathbf{I}$ where $o \in \mathcal{O}_{B \text{ import}}(\subseteq \mathcal{O})$, and the conclusion $E = \neg \square^\circ \neg A$ lies in the interval $J \in \mathbf{M}^\circ$, then it has to be the case that *the interval J lies in the interval I .*
11. $R = D$ export.
If the premiss A lies in interval $I \in \mathbf{M}^\circ$ where $o \in \mathcal{O}_{D \text{ export}}(\subseteq \mathcal{O})$, $a \in \text{People}$ and the conclusion $E = \neg \square^\circ \neg A$ lies in the interval $J \in \mathbf{I}$, then it has to be the case that *the interval I lies in the interval J .*
12. $R = T$ export.
If the premiss A lies in interval $I \in \mathbf{M}^\circ$ where $o \in \mathcal{O}_{T \text{ export}}(\subseteq \mathcal{O})$, and the conclusion $E = A$ lies in the interval $J \in \mathbf{I}$, then it has to be the case that *the interval I lies in the interval J .*

Note that for the K -rules we only demanded that the operator-index of the modality match that of the modal subordinate proof. For the other rules we also demand that the operator-index o is an element of the set of operators $\mathcal{O}_{\text{rule}}$ for which the rule is to hold, this is a convenient way of specifying different combinations of rules for different operators.

All the modal deduction rules presented here act on formulas containing only universal modal operators (\square°). To bring out the relation between axioms and deduction rules as clearly as possible, we will henceforth write all axioms using only universal operators. This means that the existential operators F and P used in section 2 will be replaced by $\neg G \neg$ and $\neg H \neg$. For a deductive treatment of existential modalities, the reader is referred to the appendix of this paper.

3.2. Natural deduction for K_t

To give a natural deduction system for K_t in the above framework, we first state that the operators of this multi-modal logic are G and H : $\mathcal{O} = \{G, H\}$. This ensures that K -import and K -export hold for both operators, allowing the derivation of the normality axioms $G(\varphi \supset \psi) \supset (G\varphi \supset G\psi)$ and $H(\varphi \supset \psi) \supset (H\varphi \supset H\psi)$. We show this for the first axiom:

| | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------|---------------|------------------|--|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|------------------|--|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----|------------------------|--|----|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------|------------------|--|----|-----------------|-------------------------|----|----------------------------------------------------------------|--|
| 1. | $G(A \supset B)$ | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 2. | <table style="border-collapse: collapse; margin-left: 5px;"> <tr> <td style="border-left: 1px solid black; padding-left: 5px;">GA</td> <td></td> </tr> </table> | GA | | | | | | | | | | | | | | | | | | | | | | | | | |
| GA | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3. | <table style="border-collapse: collapse; margin-left: 5px;"> <tr> <td style="border-left: 1px solid black; padding-left: 5px;">$G(A \supset B)$</td> <td style="padding-left: 10px;">(reiteration 1)</td> </tr> </table> | $G(A \supset B)$ | (reiteration 1) | | | | | | | | | | | | | | | | | | | | | | | | |
| $G(A \supset B)$ | (reiteration 1) | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 4. | <table style="border-collapse: collapse; margin-left: 5px;"> <tr> <td style="border-left: 1px solid black; padding-left: 5px;"> <table style="border-collapse: collapse; margin-left: 5px;"> <tr> <td style="border: 1px solid black; padding: 2px;">G</td> <td style="padding: 0 5px;">$A \supset B$</td> <td style="padding-left: 10px;">(K-import 3)</td> </tr> </table> </td> <td></td> </tr> <tr> <td style="padding-right: 10px;">5.</td> <td style="border-left: 1px solid black; padding-left: 5px;"> <table style="border-collapse: collapse; margin-left: 5px;"> <tr> <td style="border-left: 1px solid black; padding-left: 5px;">A</td> <td style="padding-left: 10px;">(K-import 2)</td> </tr> </table> </td> <td></td> </tr> <tr> <td style="padding-right: 10px;">6.</td> <td style="border-left: 1px solid black; padding-left: 5px;"> <table style="border-collapse: collapse; margin-left: 5px;"> <tr> <td style="border-left: 1px solid black; padding-left: 5px;">B</td> <td style="padding-left: 10px;">(\supset-elim 4,5)</td> </tr> </table> </td> <td></td> </tr> <tr> <td style="padding-right: 10px;">7.</td> <td style="border-left: 1px solid black; padding-left: 5px;"> <table style="border-collapse: collapse; margin-left: 5px;"> <tr> <td style="border-left: 1px solid black; padding-left: 5px;">GB</td> <td style="padding-left: 10px;">(K-export 6)</td> </tr> </table> </td> <td></td> </tr> <tr> <td style="padding-right: 10px;">8.</td> <td style="border-left: 1px solid black; padding-left: 5px;">$GA \supset GB$</td> <td style="padding-left: 10px;">(\supset-intro 2-7)</td> </tr> <tr> <td style="padding-right: 10px;">9.</td> <td colspan="2">$G(A \supset B) \supset GA \supset GB$ (\supset-intro 1-8)</td> </tr> </table> | <table style="border-collapse: collapse; margin-left: 5px;"> <tr> <td style="border: 1px solid black; padding: 2px;">G</td> <td style="padding: 0 5px;">$A \supset B$</td> <td style="padding-left: 10px;">(K-import 3)</td> </tr> </table> | G | $A \supset B$ | (K -import 3) | | 5. | <table style="border-collapse: collapse; margin-left: 5px;"> <tr> <td style="border-left: 1px solid black; padding-left: 5px;">A</td> <td style="padding-left: 10px;">(K-import 2)</td> </tr> </table> | A | (K -import 2) | | 6. | <table style="border-collapse: collapse; margin-left: 5px;"> <tr> <td style="border-left: 1px solid black; padding-left: 5px;">B</td> <td style="padding-left: 10px;">(\supset-elim 4,5)</td> </tr> </table> | B | (\supset -elim 4,5) | | 7. | <table style="border-collapse: collapse; margin-left: 5px;"> <tr> <td style="border-left: 1px solid black; padding-left: 5px;">GB</td> <td style="padding-left: 10px;">(K-export 6)</td> </tr> </table> | GB | (K -export 6) | | 8. | $GA \supset GB$ | (\supset -intro 2-7) | 9. | $G(A \supset B) \supset GA \supset GB$ (\supset -intro 1-8) | |
| <table style="border-collapse: collapse; margin-left: 5px;"> <tr> <td style="border: 1px solid black; padding: 2px;">G</td> <td style="padding: 0 5px;">$A \supset B$</td> <td style="padding-left: 10px;">(K-import 3)</td> </tr> </table> | G | $A \supset B$ | (K -import 3) | | | | | | | | | | | | | | | | | | | | | | | | |
| G | $A \supset B$ | (K -import 3) | | | | | | | | | | | | | | | | | | | | | | | | | |
| 5. | <table style="border-collapse: collapse; margin-left: 5px;"> <tr> <td style="border-left: 1px solid black; padding-left: 5px;">A</td> <td style="padding-left: 10px;">(K-import 2)</td> </tr> </table> | A | (K -import 2) | | | | | | | | | | | | | | | | | | | | | | | | |
| A | (K -import 2) | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 6. | <table style="border-collapse: collapse; margin-left: 5px;"> <tr> <td style="border-left: 1px solid black; padding-left: 5px;">B</td> <td style="padding-left: 10px;">(\supset-elim 4,5)</td> </tr> </table> | B | (\supset -elim 4,5) | | | | | | | | | | | | | | | | | | | | | | | | |
| B | (\supset -elim 4,5) | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 7. | <table style="border-collapse: collapse; margin-left: 5px;"> <tr> <td style="border-left: 1px solid black; padding-left: 5px;">GB</td> <td style="padding-left: 10px;">(K-export 6)</td> </tr> </table> | GB | (K -export 6) | | | | | | | | | | | | | | | | | | | | | | | | |
| GB | (K -export 6) | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 8. | $GA \supset GB$ | (\supset -intro 2-7) | | | | | | | | | | | | | | | | | | | | | | | | | |
| 9. | $G(A \supset B) \supset GA \supset GB$ (\supset -intro 1-8) | | | | | | | | | | | | | | | | | | | | | | | | | | |

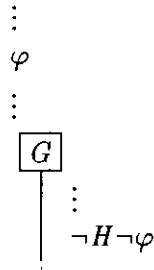
Besides *Modus Ponens*, the original definition of K_t (section 2.1) contains two inference rules called Necessitation:

Necessitation_G if φ is a thesis, then $G\varphi$ is a thesis

Necessitation_H if φ is a thesis, then $H\varphi$ is a thesis.

These inference rules are covered deductively by the K -export rules: if φ is a thesis of $K_t(\vdash \varphi)$, it can be derived without hypothesis. Hence the proof of φ is categorical, which means that after putting this proof in a G - or H -subordinate proof, K -export is applicable, resulting in a proof of $G\varphi$ or $H\varphi$. Since these proofs do not depend on further assumptions, $G\varphi$ and $H\varphi$ are then theses ($\vdash G\varphi, \vdash H\varphi$) of K_t .

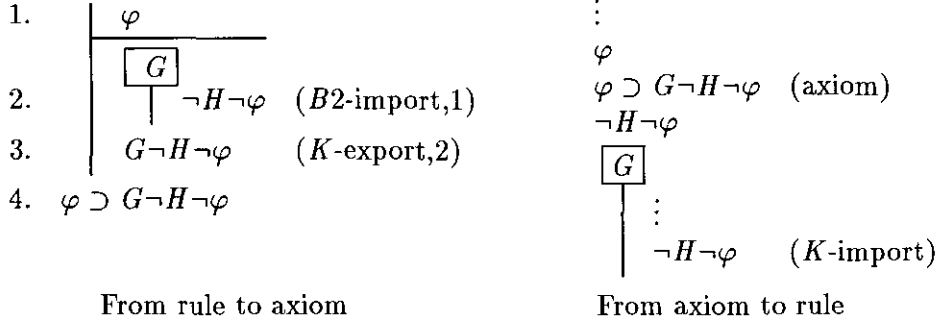
In the basic tense logic, the two operators are related by the symmetry axioms $\varphi \supset G\neg H\neg\varphi$ and $\varphi \supset H\neg G\neg\varphi$. These axioms cannot be accounted for by means of just K -import and K -export. Also none of the additional modal rules presented in the previous section directly corresponds to the symmetry axioms. However, the B -import rule provides some insight into the form of the rules that are to relate G and H : with B -import and K -export the symmetry axiom for a *single* operator ($\varphi \supset \Box^o\neg\Box^o\neg\varphi$) is derivable. With this in mind, it is not difficult to come up with rules for multi-modal symmetry, for instance for the axiom $\varphi \supset G\neg H\neg\varphi$:



The axiom expresses that if φ holds now, we should be able to see φ somewhere in the past ($\neg H\neg\varphi$) when we look back from any future time point. Hence deductively a formula

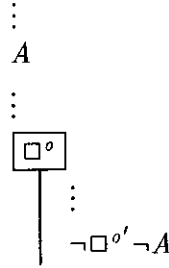
φ occurring in the main proof should be available as $\neg H\neg\varphi$ in G -subordinate proofs which represent arbitrary future time points.

Because of its resemblance to B -import, we call this rule ‘ $B2$ -import’, for ‘bi-modal B -import’. Using the normality of the G -operator, it is easy to show that the axiom can be derived by the rule and that the rule is derivable in the presence of the axiom:



Obviously, substituting G for H and H for G in the above diagrams will give us the $B2$ -import rule for the ‘mirror axiom’ $\varphi \supset H\neg G\neg\varphi$. Hence we add $B2$ -import to the deduction system in the following general form.

3.11. DEFINITION. $B2$ -import



$R = B2$ -import.

If the premiss A lies in the interval $I \in \mathbf{I}$, where $(o, o') \in \mathcal{O}_{B2\text{-import}} (\subseteq \mathcal{O} \times \mathcal{O})$, and the conclusion $E = \neg \square^{o'} \neg A$ lies in the interval $J \in \mathbf{M}^o$, then it has to be the case that the interval J lies in the interval I .

The basic tense logic K_t can now be formulated deductively as: the multi-modal Fitch-style system with $\mathcal{O} = \{G, H\}$ and $\mathcal{O}_{B2\text{-import}} = \{(G, H), (H, G)\}$.

3.3. Further properties of the tenses

There are two ways of extending the deduction system for K_t to accommodate stronger logics:

Extension by axioms

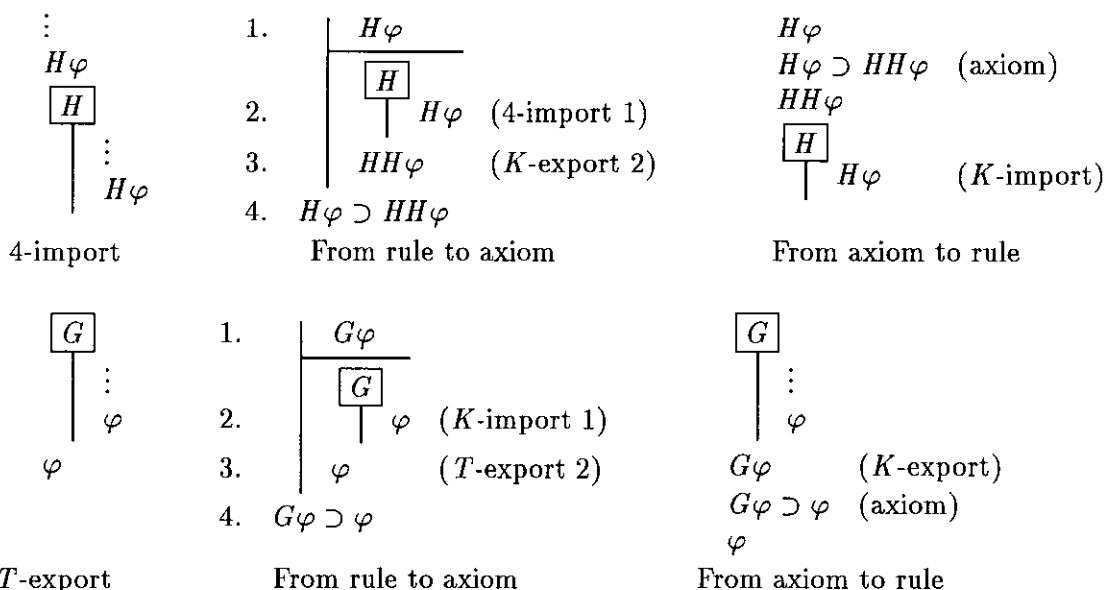
Allow tense logical axioms to be used as ‘tacit assumptions’: they may be written anywhere in a natural deduction proof without further justification. Conclusions not available in K_t can then be reached by proving the antecedent of an axiom using the K_t -rules, writing that axiom as a line in the proof and moving to the consequent of the axiom through *Modus Ponens*.

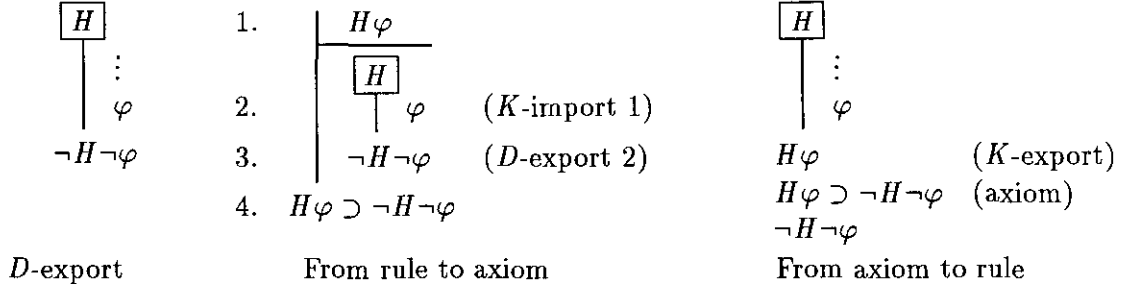
Extension by rules

Add import and export rules to K_t , allowing for more ways of transporting formulas from the main proof to the modal subordinate proof and vice versa. Conclusions not available in K_t can then be reached because: more kinds of formulas can be transferred to the modal subordinate proof to be combined there (additional import rules), and formulas derived in the subordinate proof can be brought back to the main proof in more than one form (additional export rules).

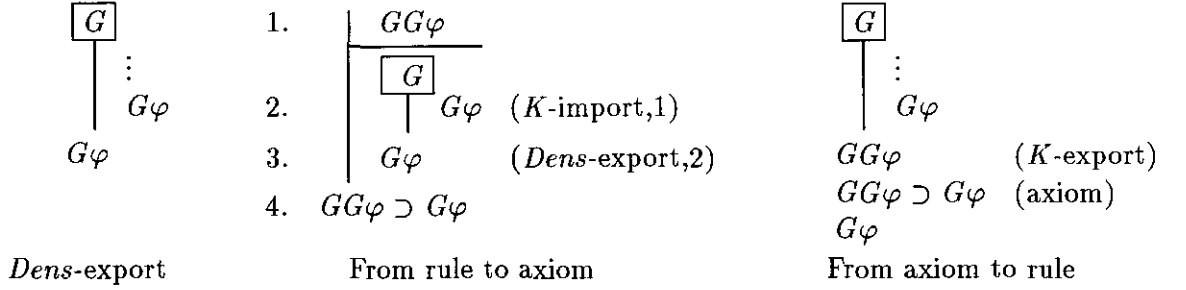
For a discussion of the relative merits of these two approaches the reader is referred to [Borghuis 1994]. Two conclusions of this discussion are relevant here. Firstly, extension by rules is preferred, because it creates a ‘separation of concerns’ in the natural deduction proofs: these proofs can be conceived of as a bundle of propositional deductions between which formulas may be exchanged. The import and export rules governing this exchange determine the tense logical strength of the system. In other words reasoning *at* a certain time point is purely propositional, whereas modal reasoning takes place in *moving between* time points. The type theoretical advantage of this separation of concerns will become clear in section 4.6. Unfortunately the second conclusion is that it seems unlikely that for every given modal axiom a corresponding import or export rule can be found, meaning that in some cases extension by axioms is the only option. This issue will be addressed at the end of the section.

For the ‘pure’ tense axioms $4_f/4_p$, T_f/T_p , and D_f/D_p we can make immediate use of the additional mono-modal import and export rules defined in the previous section. In each case we show, for one member of the axiom-pair, the rule, the derivation of the axiom by the rule, and how the rule can be derived in the presence of the axiom:



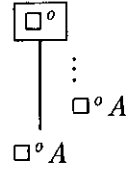


The density axioms ($GG\varphi \supset G\varphi$, $HH\varphi \supset H\varphi$) are derivable by combining *K-import* with *T-export*, but this is of no help in building a deduction system for weaker logics which extend K_t with just $Dens_f/Dens_p$. A weaker version of *T-export* is needed that corresponds directly to these axioms:



This '*Dens-export*' rule allows only formulas of the form $G\varphi$ ($H\varphi$) derived in a G -subordinate (H -subordinate) proof to be brought back to the main proof unchanged, whereas *T-export* allows this for all formulas regardless of their form.

3.12. DEFINITION. **Dens-export**



$R = Dens\ export$.

If the premiss $\square^o A$ lies in interval $I \in \mathbf{M}^o$ where $o \in \mathcal{O}_{Dens\ export} (\subseteq \mathcal{O})$, and the conclusion $E = \square^o A$ lies in the interval $J \in \mathbf{I}$, then it has to be the case that *the interval I lies in the interval J*.

Regarding the 'mixed' axioms, extensions by rules can be given for the preservation axioms ($\neg H \neg \varphi \supset G \neg H \neg \varphi$, $\neg G \neg \varphi \supset H \neg G \neg \varphi$) and the linearity axioms ($\blacksquare \varphi \supset GH\varphi$, $\blacksquare \varphi \supset HG\varphi$) but we postpone this to sections 3.4 and 3.5 that deal with the influence of symmetry on extensions of K_t and with linear tense logics respectively. This leaves the following axioms of the lists presented in section 2.2:

$$Z_f \quad G(G\varphi \supset \varphi) \supset (\neg G \neg G\varphi \supset G\varphi)$$

$$Z_p \quad H(H\varphi \supset \varphi) \supset (\neg H \neg H\varphi \supset H\varphi)$$

$$W_f \quad G(G\varphi \supset \varphi) \supset G\varphi$$

$$W_p \quad H(H\varphi \supset \varphi) \supset \varphi$$

$$\begin{array}{l}
Dum_f \quad G(G(\varphi \supset G\varphi) \supset \varphi) \supset (\neg G\neg G\varphi \supset \varphi) \quad Dum_p \quad H(H(\varphi \supset H\varphi) \supset \varphi) \supset (\neg H\neg H\varphi \supset \varphi) \\
Cont \quad \blacksquare(H\varphi \supset \neg G\neg H\varphi) \supset (H\varphi \supset G\varphi)
\end{array}$$

For these axioms no rules of the above kind, i.e. rules which transfer a formula between proofs changing just the modality of that formula (its main connective), can be found. Rules for Z_f/Z_p (export) and W_f/W_p (import) do suggest themselves, but these rules change the matrix of a formula whilst transferring it. Such rules raise questions about the expressive limitations of Fitch-style modal deduction and since we want to discuss these questions against the background of the interpretation of the Fitch-style proofs in type theory, we postpone this discussion till section 5. Meanwhile we resort to extension by axioms for Z_f/Z_p , W_f/W_p , Dum_f/Dum_p , and $Cont$.

3.4. The interaction between past and future

In the minimal tense logic K_t , the descriptions of the future and past directions of time are not independent. The operators ‘ G ’ and ‘ H ’ are related by the symmetry axioms that express the intuition that the flow of time is ‘isotropic’; the observable properties of the ordering of the time points in future and past direction are the same. In this section we look at two ways in which the influence of this basic symmetry extends beyond K_t .

3.4.1. The mirror image property

In the presentation of K_t (section 2.1), the ‘mirror image property’ of this logic was pointed out: if a formula $\varphi(G, H)$ is a theorem of K_t , so is $\varphi(H, G)$. This means that given a theorem, we can obtain a new theorem ‘for free’ by simply changing all occurrences of G into occurrences of H and all occurrences of H into occurrences of G . Rephrased for the natural deduction system for K_t , the mirror image property states that given a natural deduction proof for $\varphi(G, H)$ we should be able to find a natural deduction proof for $\varphi(H, G)$. It is not difficult to see why this property should hold for the K_t -deduction system: for both ‘ G ’ and ‘ H ’ the system has K -import and K -export, and the $B2$ -import rules that relate the operators are each other’s ‘mirror image’ (one turns an occurrence of φ into an occurrence of $\neg G\neg\varphi$ inside an H -subordinate proof, the other turns φ into $\neg H\neg\varphi$ inside a G -subordinate proof). Therefore we can find a proof for $\varphi(H, G)$, given a proof for $\varphi(G, H)$, by taking the mirror image of all hypotheses occurring in the original proof and then matching all rule applications in the original proof step by step with applications of the mirror images of those rules in the new proof.

In general the mirror image property is not preserved when the logic K_t is extended with further axioms, but there are cases in which it is: sometimes adding an axiom expressing a property of one direction of time will automatically result in the validity of the mirror image of that axiom. To see this, we go back to the model theoretic effect of the symmetry axioms: in any model of K_t , given two time points s and t $sR_G t \Leftrightarrow sR_H t$ (where $sR_G t = tR_H s$). Hence adding an axiom to K_t that corresponds to a relational property that is invariant under ‘reversing the arrows’ can yield the validity of mirror images. A trivial example of this is extending K_t with the T -axiom for one direction of time, say $G\varphi \supset \varphi$. Since this axiom corresponds to reflexivity of the R_G relation, $tR_G t$ will hold for all time points in all models. Because of the symmetry $tR_H t$ also holds for all time points in all models, and so $H\varphi \supset \varphi$ will be a theorem of $K_t + G\varphi \supset \varphi$ without having been added to K_t explicitly. In natural

deduction, this example takes the following form: adding T -export for G to the deduction system for K_t makes $H\varphi \supset \varphi$ derivable (by means of a simple proof*).

Although a syntactic characterization of relational properties that are preserved under reversing the arrows is still to be found [van Benthem 1983], a number of interesting cases has been noted in the literature. In this section we look at two of these (one involving an export-rule and one involving an import-rule), to show how the modal rules for symmetry in K_t combine with the rule for an additional axiom in the deduction proof for the mirror image of that axiom.

EXAMPLE 1.

The first case is the extension of K_t with the 4-axioms. These axioms ($G\varphi \supset GG\varphi/H\varphi \supset HH\varphi$) correspond to transitivity of the accessibility relations (R_G and R_H respectively) in the models of $K_t + 4$. Assuming that we add just the 4_G -axiom ($G\varphi \supset GG\varphi$) to K_t , all models of the extended logic will have that for any time points s, t, u : $sR_G t \& tR_G u \Rightarrow sR_G u$. Reversing the relations in this property yields $tR_G s \& uR_G t \Rightarrow uR_G s$, which is equivalent to $sR_H t \& tR_H u \Rightarrow sR_H u$, hence R_H is also transitive in all models and so $H\varphi \supset HH\varphi$ is a theorem of $K_t + 4_G$. To show the mirror image property of the 4-axiom deductively, we extend the natural deduction system for K_t with the 4-import rule for G after which $H\varphi \supset HH\varphi$ can be proved as follows:

| | | |
|-----|-----------------------------------------------------------------------------------------------------|--------------------------|
| 1. | $H\varphi$ | |
| | <div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">H</div> | |
| 2. | $\neg G\neg H\varphi$ | (B2-import 1) |
| | <div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">H</div> | |
| 3. | $\neg G(\neg\neg)G\neg H\varphi$ | (B2-import 2) |
| 4. | $\neg\varphi$ | |
| | <div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">G</div> | |
| 5. | $\neg H(\neg\neg)\varphi$ | (B2-import 4) |
| 6. | $G\neg H\varphi$ | (K-export 5) |
| | <div style="border: 1px solid black; display: inline-block; padding: 2px 5px;">G</div> | |
| 7. | $G\neg H\varphi$ | (4-import 6) |
| 8. | $GG\neg H\varphi$ | (K-export 7) |
| 9. | $\neg GG\neg H\varphi$ | (reiteration 3) |
| 10. | $\neg\neg\varphi$ | (\neg -intro 4-9) |
| 11. | φ | (\neg -elim 10) |
| 12. | $H\varphi$ | (K-export 11) |
| 13. | $HH\varphi$ | (K-export 12) |
| 14. | $H\varphi \supset HH\varphi$ | (\supset -intro 1-11) |

Note that the derivation of $\neg GG\neg H\varphi$ from $\neg G\neg\neg G\neg H\varphi$ (between lines 3 and 9) was left

*A hint for readers who want to try this: assume $H\varphi$ and $\neg\varphi$, and derive a contradiction using B2-import and T -export for G .

out in this proof, as well as the derivation of $\neg H\neg\neg\varphi$ (between lines 5 and 6) were left out. Since the removal of embedded double negations is a routine task (as the reader can easily check for himself), we merely indicated the presence of these double negations between brackets but did not spell out their removal. This ‘abbreviation’ will be used in deduction proofs throughout the paper.

EXAMPLE 2.

When extending K_t with the *Dens*-axioms ($GG\varphi \supset G\varphi / HH\varphi \supset H\varphi$), it is sufficient to add one of them to K_t to turn both axioms into theorems of the extended logic. These axioms enforce the property of density for the accessibility relations in the models, hence if we add $HH\varphi \supset H\varphi$ to K_t , a third time point can be found between any two time points related by R_H in a model: $sR_H t \Rightarrow \exists u sR_H u \& uR_H t$. Reversing the relations yields: $tR_H s \Rightarrow \exists u uR_H s \& tR_H u$. Hence, by the definition $sR_G t = tR_H s$, we have density for R_G and so $GG\varphi \supset G\varphi$ is a theorem of $K_t + \text{Dens}_H$.

Deductively, the mirror image property of the *Dens*-axioms is shown by extending the deduction system for K_t with *Dens*-export for H and proving $GG\varphi \supset G\varphi$:

| | | |
|-----|------------------------------|--------------------------|
| 1. | $GG\varphi$ | |
| | \boxed{G} | |
| 2. | $\neg H\neg GG\varphi$ | (B2-import 1) |
| 3. | $H\neg G\varphi$ | |
| | \boxed{H} | |
| 4. | $\neg G\varphi$ | (K-import 3) |
| | \boxed{H} | |
| 5. | $\neg G(\neg\neg)G\varphi$ | (B2-import 4) |
| 6. | $H\neg GG\varphi$ | (K-export 5) |
| 7. | $H\neg GG\varphi$ | <i>Dens</i> -export 6) |
| 8. | $\neg H\neg GG\varphi$ | (reiteration 2) |
| 9. | $\neg H\neg G\varphi$ | (\neg -intro 3-8) |
| 10. | $\neg\varphi$ | |
| | \boxed{H} | |
| 11. | $\neg G(\neg\neg)\varphi$ | (B2-import 10) |
| 12. | $H\neg G\varphi$ | (K-export 11) |
| 13. | $\neg H\neg G\varphi$ | (reiteration 9) |
| 14. | $\neg\neg\varphi$ | (\neg -intro 10-13) |
| 15. | φ | (\neg -elim 14) |
| 16. | $G\varphi$ | (K-export 15) |
| 17. | $GG\varphi \supset G\varphi$ | (\supset -intro 1-15) |

These examples show that the Fitch-style treatment of symmetry in K_t by means of the

$B2$ -import rules correctly accounts for the preservation of the mirror image property: in extending K_t , it is sufficient to add just one of the 4-import rules or $Dens$ -export rules to obtain proofs of both 4-axioms or $Dens$ -axioms.

Since the ‘mirror axiom’ becomes derivable after adding the rule for one operator, the rule for the other operator is redundant: it is derivable in the presence of the axiom. However, the examples also show that the deduction proofs for the mirror axioms are not always obvious or easy to find. Therefore it makes sense to include both rules in the deduction system for the extension of K , if we are after a practical rather than a ‘minimal’ deduction system. This is the strategy we apply in extending K_t with 4 and $Dens$ and in similar cases throughout the rest of this paper.

3.4.2. General interaction patterns

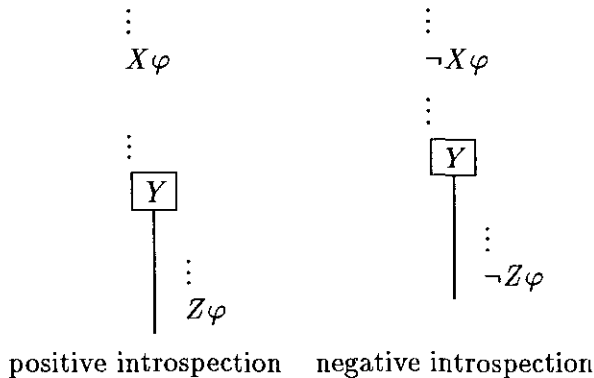
A second effect of the symmetry between G and H is that it can cause dependencies between different extensions of K_t , which would have been independent had the operators been unrelated. As with the preservation of the mirror image property, there is no systematic account of these phenomena but we look at a couple of cases to see how extensions of K_t can interact in the natural deduction system.

To structure the discussion, we invoke a classification of axioms involving multiple O from [van der Hoek 1992]. This classification allows us to find the import or export rules corresponding to these axioms in a somewhat systematic way ([Borghuis 1994]). In the following the letters X , Y and Z range over normal modal operators (i.e. G and H):

- a) $X\varphi \supset YZ\varphi$ are called positive introspection (pi. -) formulas
- b) $\neg X\varphi \supset Y\neg Z\varphi$ are called negative introspection (ni. -) formulas
- c) $XY\varphi \supset Z\varphi$ are called positive extraspection (pe. -) formulas
- d) $X\neg Y\varphi \supset \neg Z\varphi$ are called negative extraspection (ne. -) formulas.

Instantiations of a) – d) are collectively referred to as *introspection formulas*.

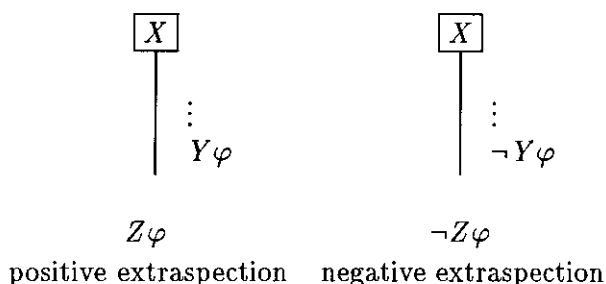
Given only that X , Y , and Z are normal operators, the introspection formulas correspond to import rules and the extraspection formulas to export rules:



positive introspection

If the premiss $X\varphi$ lies in the interval $I \in \mathbf{I}$ and the conclusion $E = Z\varphi$ lies in the interval $J \in \mathbf{M}^Y$, then it has to be the case that *the interval J lies in the interval I* . **negative introspection**

If the premiss $\neg X\varphi$ lies in the interval $I \in \mathbf{I}$ and the conclusion $E = \neg Z\varphi$ lies in the interval $J \in \mathbf{M}^Y$, then it has to be the case that *the interval J lies in the interval I* .



positive extraspection

If the premiss $Y\varphi$ lies in interval $I \in \mathbf{M}^X$ and the conclusion $E = Z\varphi$ lies in the interval $J \in \mathbf{I}$, then it has to be the case that *the interval I lies in the interval J* .

negative extraspection

If the premiss $\neg Y\varphi$ lies in interval $I \in \mathbf{M}^X$ and the conclusion $E = \neg Z\varphi$ lies in the interval $J \in \mathbf{I}$, then it has to be the case that *the interval I lies in the interval J* .

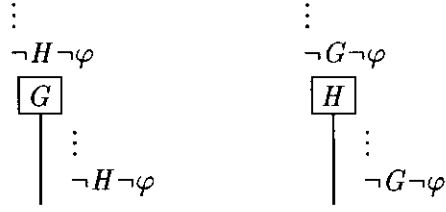
All modal rules above K -import and K -export fit into this classification if we allow modalities to be identified (e.g. $X = Y$) and ‘left out’; replaced with the ‘empty modal operator’ denoted by ‘ \emptyset ’ (for instance $X = \emptyset$):

- | | |
|-------------------------------|-------------------------------------------------------------------------------------------------------------|
| Positive introspection | 4-import: $X = Y = Z$ |
| Negative introspection | 5-import: $X = Y = Z$ B -import: $X = \emptyset, Y = Z^*$ $B2$ -import: $X = \emptyset, Y \neq Z^*$ |
| Positive extraspection | $Dens$ -export: $X = Y = Z$ T -export: $X = Z, Y = \emptyset$ |
| Negative extraspection | D -export: $X = Z, Y = \emptyset$.* |

In the next section, which deals with the deductive treatment of linear temporal flows, rules will be proposed which use the full generality of the schemata in the sense that they involve three different modal operators.

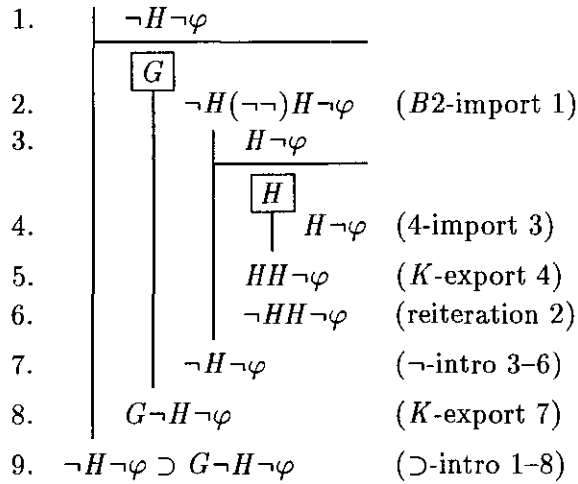
Using the classification, it is easy to find the rules corresponding to the last pair of interaction axioms discussed in 2.2.2, the ‘preservation axioms’: $\neg H\neg\varphi \supset G\neg H\neg\varphi$ and $\neg G\neg\varphi \supset H\neg G\neg\varphi$. These are clearly cases of negative introspection (with $X = Z = H, Y = G$ and $X = Z = G, Y = H$ respectively), which give rise to the following import rules:

* Where the occurrences of ‘ φ ’ are replaced by occurrences of ‘ $\neg\varphi$ ’ in the rule schemas, yielding ‘ $\neg\neg\varphi$ ’ in some cases which is to be replaced by the equivalent ‘ φ ’.



In keeping with the nomenclature used so far, these negative introspection cases where $X = Z \neq Y$ are called ‘52-import’ rules (cf. *B2-import* versus *B-import*).

In multi-modal logics where there are no pairs of symmetrically related operators (like in the epistemic/doxastic logics from which the above classification originates, see [van der Hoek 1992]), extensions of the basic logic K with positive introspection axioms are independent of extensions with negative introspection axioms. For the basic tense logic K_t such extensions are not independent: in $K_t + G\varphi \supset GG\varphi/H\varphi \supset HH\varphi$ (positive introspection) the preservation axioms $\neg H\neg\varphi \supset G\neg H\neg\varphi/\neg G\neg\varphi \supset H\neg G\neg\varphi$ (negative introspection) are theorems. Deductively this means that the preservation axioms should be derivable in the natural deduction system for K_t extended with the 4-import rules. We show this for $\neg H\neg\varphi \supset G\neg H\neg\varphi$.



Using the ‘mirror images’ of *B2-import* and 4-import, the other preservation axiom ($\neg G\neg\varphi \supset H\neg G\neg\varphi$) is derivable in exactly the same way. Hence the 52-import rules for these axioms proposed earlier are derived rules of the deduction system for $K_t + 4$. This does not come as a complete surprise, since the symmetry axioms are themselves (simple) cases of negative introspection. However, the interaction between symmetry and transitivity is not trivial: the proofs of the preservation axioms can be generalized to the following proposition relating the categories of positive and negative introspection in the classification of multi-modal axioms.

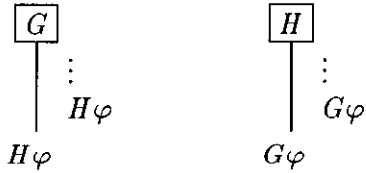
PROPOSITION.

A negative introspection formula $\neg X\neg\varphi \supset Y\neg X\neg\varphi$ is derivable for normal operators X , Y and Z where Y and Z are related by symmetry, $\varphi \supset Y\neg Z\neg\varphi$ (negative introspection where $X = \emptyset$), and X and Z by positive introspection, $X\varphi \supset ZX\varphi$.

PROOF.

| | | |
|----|----------------------------------------|----------------------------|
| 1. | $\neg X\varphi$ | |
| | \boxed{Y} | |
| 2. | $\neg Z(\neg\neg)X\varphi$ | (symmetry 1) |
| 3. | $X\varphi$ | |
| | \boxed{Z} | |
| 4. | $X\varphi$ | (positive introspection 3) |
| 5. | $ZX\varphi$ | (K-export 4) |
| 6. | $\neg ZX\varphi$ | (reiteration 2) |
| 7. | $\neg X\varphi$ | (\neg -intro 3-6) |
| 8. | $Y\neg X\varphi$ | (K-export 7) |
| 9. | $\neg X\varphi \supset Y\neg X\varphi$ | (\supset -intro 1-8) |

The symmetry between the operators can also influence the extension of K_t with extraspection rules, for instance in the case of the ‘actuality axioms’: $GH\varphi \supset H\varphi$ and $HG\varphi \supset G\varphi$. These axioms state that if always in one direction of time something holds always in the other direction of time, it already holds for that other direction of time at this actual moment. By themselves the axioms are clearly cases of positive extraspection, giving rise to export rules like those for the density axioms:



However, the actuality axioms are also a theorem of $K_t + 4$ extended with seriality ($G\varphi \supset \neg G\neg\varphi$, $H\varphi \supset \neg H\neg\varphi$). We show the case for $GH\varphi \supset H\varphi$:

| | | |
|-----|------------------------------|--------------------------|
| 1. | $GH\varphi$ | |
| 2. | $\neg H\varphi$ | |
| 3. | \boxed{G} | |
| 4. | $\neg H(\neg\neg)H\varphi$ | (B2-import 2) |
| 5. | $H\varphi$ | |
| 6. | \boxed{H} | |
| 7. | $H\varphi$ | (4-import 4) |
| 8. | $HH\varphi$ | (K-export 5) |
| 9. | $\neg HH\varphi$ | (reiteration 3) |
| 10. | $\neg H\varphi$ | (\neg -intro 4-7) |
| 11. | $\neg G(\neg\neg)H\varphi$ | (D-export 8) |
| 12. | $GH\varphi$ | (reiteration 1) |
| 13. | $\neg\neg H\varphi$ | (\neg -intro 2-10) |
| | $H\varphi$ | (\neg -elim 11) |
| | $GH\varphi \supset H\varphi$ | (\supset -intro 1-12) |

Hence the export rules for the actuality axioms proposed above are derived rules in the deduction system for K_t+4+D . In the same way as for the preservation axiom, this connection between extensions with axioms of different classes can be formulated in a more general way: A positive extraspection formula $XY\varphi \supset Y\varphi$ is derivable for normal modal operators X , Y , and Z where Y and Z are related by symmetry, $\varphi \supset Y\neg Z\neg\varphi$, and positive introspection $Y\varphi \supset ZY\varphi$, and negative extraspection $X\varphi \supset \neg X\neg\varphi$ holds for X .

As stated before, there are no general results about the influence of symmetry on the extensions of K_t in the literature. The examples discussed in this section show that cases that have been noted in model theory or axiomatics have a counterpart in the Fitch-style deduction systems. This is all we can hope to achieve at the moment, even working out all possible dependencies between axiom classes in the classification of inspection formulas will not provide us with a general picture of the influence of symmetry since the classification covers only part of even the most well-known temporal axioms.

3.5. Linear tense logics

Armed with the basic deduction system for K_t and the extensions discussed in the previous section, we will now attempt to give a deductive formulation of the tense logics for linear flows of time that were introduced in section 2.3.

3.5.1. The logic *Lin*

The logic of all linear flows of time, *Lin*, was defined as K_t plus the axioms for transitivity and linearity of past and future:

$$\begin{array}{ll}
4_f \ G\varphi \supset GG\varphi & 4_p \ H\varphi \supset HH\varphi \\
L_f \ \blacksquare\varphi \supset HG\varphi & L_p \ \blacksquare\varphi \supset GH\varphi
\end{array}$$

The 4-axioms can be dealt with straightforwardly by adding the 4-import rule to K_t for both the ‘ G ’ and the ‘ H ’ operator ($\mathcal{O}_{4import} = \{G, H\}$). For the linearity axioms L_f and L_p we can go in two directions, depending on the treatment of the universal operator ‘ \blacksquare ’:

1. View the universal operator as a mere abbreviation of a complex formula: $\blacksquare\varphi =_{def} H\varphi \wedge \varphi \wedge G\varphi$. The deduction system has no rules for the operator; it allows us to abbreviate formulas using ‘ \blacksquare ’, but ‘calculates’ with the complex formulas using the rules for ‘ G ’, ‘ H ’, and ‘ \wedge ’.
2. View ‘ \blacksquare ’ as an operator in its own right, giving modal rules for it in the deduction system.

Initially, option 1 seems to be preferable; the definition $\blacksquare\varphi = H\varphi \wedge \varphi \wedge G\varphi$ shows that the universal operator does not introduce something conceptually new in the deduction system. This idea is confirmed by the easy derivations of the formulas corresponding to the abbreviations:

- $\blacksquare(\varphi \supset \psi) \supset (\blacksquare\varphi \supset \blacksquare\psi)$ (normality)
- $\blacksquare\varphi \supset \blacksquare\blacksquare\varphi$ (transitivity)
- $\blacksquare\varphi \supset \varphi$ (reflexivity)
- $\blacksquare\varphi \supset G\varphi, \blacksquare\varphi \supset H\varphi$ (\blacksquare is stronger than G and H)

However, it is not so clear what additional modal rules would allow us to derive the linearity axioms in unabbreviated form (given the standard rules for ‘ \wedge ’, ‘ G ’, and ‘ H ’). There seem to be no import- or export-rules comparable to the ones we encountered so far, that would make $(H\varphi \wedge \varphi \wedge G\varphi) \supset HG\varphi$ (L_f) or $(H\varphi \wedge \varphi \wedge G\varphi) \supset GH\varphi$ (L_p) derivable.

Under option 2 we can find natural deduction rules corresponding to the linearity axioms, but we have to do more work because the relation between ‘ \blacksquare ’ and the operators ‘ G ’ and ‘ H ’ is not given in the deduction system. To capture this relation we need deduction rules corresponding to the principles in the above list. The first step is to add the universal modality to the set of operators: $\mathcal{O} = \{G, H, \blacksquare\}$. In this way we automatically get K -import and K -export for ‘ \blacksquare ’, making normality for this operator derivable. For the other properties listed above further rules must be added: 4-import for transitivity, T -export for reflexivity, and less traditional rules for the principles

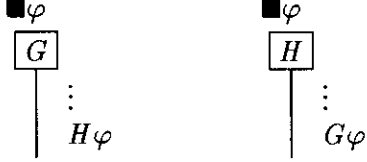
$$\blacksquare\varphi \supset G\varphi, \blacksquare\varphi \supset H\varphi$$

$$\blacksquare\varphi \supset GH\varphi \text{ (} L_p \text{)}, \blacksquare\varphi \supset HG\varphi \text{ (} L_f \text{)}.$$

Fortunately, all of these axioms fit in the classification of interactions discussed above as ‘positive introspection formulas’. This allows us to give rules for the axioms by simply substituting the relevant operators for ‘ X ’, ‘ Y ’, and ‘ Z ’ in the rule-schema for positive introspection. For the axioms $\blacksquare\varphi \supset G\varphi$ and $\blacksquare\varphi \supset H\varphi$, these substitutions are $X := \blacksquare, Y = Z = G$, and $X := \blacksquare, Y = Z = H$ respectively, which give rise to the following import rules:



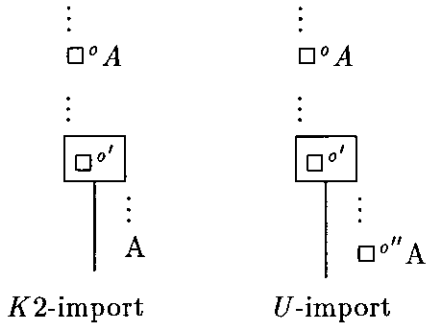
Rules of this sort occur in a different setting in [Borghuis 1994], and shall be called ‘K2-import’ for ‘bi-modal K-import’. For the linearity axioms $\blacksquare\varphi \supset GH\varphi$ and $\blacksquare\varphi \supset HG\varphi$ the substitutions are $X := \blacksquare$, $Y := G$, $Z := H$, and $X := \blacksquare$, $Y := H$, $Z := G$, resulting in import rules:



We shall call these import rules ‘U-import’, for ‘universal’ import, since they involve the universal operator and represent the most general case of positive introspection with $X \neq Y \neq Z \neq X$.

3.13. DEFINITION. Deduction rules

These interactions can be brought into the deduction-system by adding the following rules:



3.14. DEFINITION. Application of deduction rules

Given a proof figure \mathbf{D} , with interval $D = [1, n]$, formulas F_1, \dots, F_n and intervals \mathbf{I} . A formula E is the result of an *application* of deduction rule R , if E is the conclusion of R , the premisses of R precede E , and one of the following conditions is met for the modal rules:

13. $R = K2\text{ import}$.

If the premiss $\blacksquare^o A$ lies in interval $I \in \mathbf{I}$ and the conclusion $E = A$ lies in the interval $J \in \mathbf{M}^{o'}$ where $(o, o') \in \mathcal{O}_{K2} (\subseteq \mathcal{O} \times \mathcal{O})$, then it has to be the case that *the interval J lies in the interval I* .

14. $R = U\text{ import}$.

If the premiss $\blacksquare^o A$ lies in interval $I \in \mathbf{I}$ and the conclusion $E = \blacksquare^{o''} A$ lies in the interval $J \in \mathbf{M}^{o'}$ where $(o, o', o'') \in \mathcal{O}_U (\subseteq \mathcal{O} \times \mathcal{O} \times \mathcal{O})$, then it has to be the case that *the interval J lies in the interval I* .

It would seem that adding these last two rules completes our deductive characterization of *Lin* under option 2: the axioms L_f and L_p , and all listed axioms relating ‘ \blacksquare ’ to ‘ G ’ and ‘ H ’ have become derivable. However, the proposed deductions rules cover only half of the

There they were called ‘FK-import’ for ‘forced’ K-import, but in this paper we want to avoid confusion with the tense operator ‘ F ’.

definition $\blacksquare\varphi = H\varphi \wedge \varphi \wedge G\varphi$, i.e. the *elimination* of formulas $\blacksquare\varphi$ ($\blacksquare\varphi \supset (H\varphi \wedge \varphi \wedge G\varphi)$, derivable by means of $K2$ -import and T -export). The other half of the definition, $(H\varphi \wedge \varphi \wedge G\varphi) \supset \blacksquare\varphi$, is not derivable, indicating that we have not yet covered all possibilities for *introducing* formulas of the form $\blacksquare\varphi$. By means of the K -rules we can introduce formulas of this form on the basis of other \blacksquare -formulas, but we cannot account for cases where φ holds for different reasons in different directions of time: if for instance $G(\psi \supset \varphi)$ and $G\psi$, and $(\psi \supset \varphi)$ and ψ yield φ for the all future time points and the present ($G\varphi \wedge \varphi$), but $H(\xi \supset \varphi)$ and $H\xi$ yield φ for all time points in the past ($H\varphi$), we can conclude to $\blacksquare\varphi$ by means of the definition since we have $(H\varphi \wedge \varphi \wedge G\varphi)$, but we cannot derive it by means of the deduction rules. Since no reasonable Fitch-style rule corresponding to the ‘introduction-half’ of the definition exists (this will be discussed in section 5), we will have to add it as an axiom.

Given these considerations, we can now define the deduction system for *Lin* as a multi-modal Fitch-style deduction system where:

$$\begin{aligned} \mathcal{O} &= \{G, H, \blacksquare\}, \\ \mathcal{O}_{B2\text{-import}} &= \{(G, H), (H, G)\}, \\ \mathcal{O}_{A\text{-import}} &= \{G, H, \blacksquare\}, \\ \mathcal{O}_{K2\text{-import}} &= \{(\blacksquare, G), (\blacksquare, H)\}, \\ \mathcal{O}_{U\text{-import}} &= \{(\blacksquare, G, H), (\blacksquare, H, G)\}, \\ \mathcal{O}_{T\text{-export}} &= \{\blacksquare\}, \end{aligned}$$

and we have the axiom $\blacksquare\text{-int}$ $(H\varphi \wedge \varphi \wedge G\varphi) \supset \blacksquare\varphi$.

3.5.2. Familiar linear structures

For each of the logics corresponding to a familiar linear structure (as presented in section 2.3.2), a Fitch-style deduction system can now be defined by extending the deductive system for the logic *Lin* with the appropriate import- or export-rules and axioms:

- $(\mathbb{Z}, <)$, the deduction system for *Lin* and
 - additional rules: D -export for G
 D -export for H
 - additional axioms: Z_f $G(G\varphi \supset \varphi) \supset (\neg G\neg G\varphi \supset G\varphi)$
 Z_p $H(H\varphi \supset \varphi) \supset (\neg H\neg H\varphi \supset H\varphi)$
- $(\mathbb{N}, <)$, the deduction system for *Lin* and
 - additional rules: D -export for G
 - additional axioms: Z_f $G(G\varphi \supset \varphi) \supset (\neg G\neg G\varphi \supset G\varphi)$
 W_p $H(H\varphi \supset \varphi) \supset H\varphi$
- $(\mathbb{Q}, <)$, the deduction system for *Lin* and

The same problem occurs with other operators that are model theoretically defined as the transitive reflexive closure of the accessibility relations of the operators in \mathcal{O} , such as the operator for ‘Common Knowledge’ (see [Borghuis 1994]).

additional rules: D -export for G
 D -export for H
 $Dens$ -export for G
 $Dens$ -export for H

- (Q, \leq) , the deduction system for Lin and
 additional rules: T -export for G
 T -export for H

- $(R, <)$, the deduction system for Lin and
 additional rules: D -export for G
 D -export for H ($\mathcal{O}_{Dexport} = \{G, H\}$)
 $Dens$ -export for G
 $Dens$ -export for H
 additional axiom: $Cont (H\varphi \supset \neg G \neg H\varphi) \supset (H\varphi \supset G\varphi)$

4. Tense MPTSs

The possibilities for temporal reasoning offered by the tense logics can be brought to typed λ -calculus by interpreting these logics in so-called Modal Pure Type Systems in a propositions-as-types way. After the definition of MPTSs and a brief introduction to their relation to modal logic, we show how MPTSs corresponding to the various tense logics discussed so far can be build. At the end of this section, we point out an interesting consequence of the ‘proofs-as-objects’ correspondence between modal natural deduction proofs and MPTS-terms: given a deduction proof for a certain proposition, a simpler proof of that proposition can sometimes be found by means of reductions on the MPTS-term corresponding to the deduction proof.

4.1. MPTSs with multiple modalities

MPTSs are an extension of the Pure Type Systems of [Barendregt 1992], which give a general description of a large class of typed lambda calculi providing possibilities for generic proofs of meta theoretical properties. The interpretation of (non-modal) propositional and predicate logics are well-understood, see [Geuvers 1993], which makes PTSs an excellent starting point for the construction of modal type systems. In the definition of MPTSs below, we assume that the reader is familiar with PTSs and with the propositions-as-types interpretation of propositional logic. This allows us to concentrate on the aspects that are specific for the modal systems, for a more gentle introduction the reader is referred to [Borghuis 1994].

We start the definition in the usual way, by specifying the set of pseudoterms given the set of ‘sorts’ \mathcal{S} supplemented with a set of ‘modalities’ \mathcal{O} .

4.1. DEFINITION. Pseudoterms

The set of pseudoterms \mathcal{T} over \mathcal{S} and \mathcal{O} is:

$$\mathcal{T} ::= \mathcal{S} \mid \text{Var} \mid (\Pi \text{Var} : \mathcal{T}. \mathcal{T}) \mid (\lambda \text{Var} : \mathcal{T}. \mathcal{T}) \mid \mathcal{T} \mathcal{T} \mid \Box^{\mathcal{O}} \mathcal{T} \mid \hat{k}^{\mathcal{O}} \mathcal{T} \mid \hat{k}^{\mathcal{O}} \mathcal{T} \mid \mathcal{C}$$

where Var is a countable set of variables, and \mathcal{C} is countable set of constants which will be used to deal with ‘logical axioms’.

Hence the pseudoterms are those of PTSs, complemented with ‘modal types’ ($\Box^{\mathcal{O}} \mathcal{T}$), and proof terms for the modal rules ($\hat{k}^{\mathcal{O}} \mathcal{T}$, $\hat{k}^{\mathcal{O}} \mathcal{T}$). MPTSs also have an extended set of pseudo contexts.

4.2. DEFINITION. Contexts

- (i) A *declaration* is a judgement of the form $x : A$, where x is a variable and A a pseudoterm.
- (ii) A *pseudo-context* is a finite ordered sequence of declarations $(x : A)$, all with distinct subjects: $x_1 : A_1, \dots, x_n : A_n$.
- (iii) A *generalized pseudo-context* is a finite ordered sequence of pseudo-contexts and indexed separators: $\mathcal{G} = \Gamma_1 \boxtimes^o \dots \boxtimes^{o'} \Gamma_n$ with $o, \dots, o' \in \mathcal{O}$.

Clauses (i) and (ii) define PTS-contexts, clause (iii) allows us to insert ‘separators’ (\boxtimes^o) in these contexts. The separators let us partition the declaration in the context in the same way in which modal subordinate proofs in Fitch-style deduction partition the set of hypotheses during the proof. As will become clear below, this additional structuring of the context opens

up the possibility to define type theoretical analogons of the Fitch-style import and export rules.

Given the definitions of pseudoterms and generalized contexts, and the notational abbreviation $\mathcal{G} \vdash A : B : C$ for $\mathcal{G} \vdash A : B$ and $\mathcal{G} \vdash B : C$, the derivation rules of MPTSs can be stated in the following way.

4.3. DEFINITION. Multi-Modal Pure Type Systems

A multi-modal Pure Type System with β -conversion, $\text{MPTS}_\beta^{\mathcal{O}}$, is given by a set \mathcal{S} of sorts containing *Prop*, *Set*, and *Type*, a set $\mathcal{A}^{\text{Type}} \subset \mathcal{S} \times \mathcal{S}$ of *typing axioms*, a set $\mathcal{A}^{\text{Logic}} \subset \mathcal{C} \times \mathcal{T}$ of *logical axioms*, and a set $\mathcal{R} \subset \mathcal{S} \times \mathcal{S} \times \mathcal{S}$ of *rules*. The MPTS that is given by \mathcal{S} , \mathcal{A} and \mathcal{R} is denoted by $\square\lambda_\beta(\mathcal{S}, \mathcal{A}, \mathcal{R})$ and is the typed λ -calculus with the following deduction rules:

$$\begin{array}{l}
(\text{axioms}) \quad \varepsilon \vdash s_1 : s_2 \quad \text{if } s_1 : s_2 \in \mathcal{A}^{\text{Type}} \quad \varepsilon \vdash c : A : \text{Prop} \quad \text{if } c : A \in \mathcal{A}^{\text{Logic}} \\
\\
(\text{start}) \quad \frac{\mathcal{G} \vdash A : s}{\mathcal{G}, x : A \vdash x : A} \\
\\
(\text{weakening}) \quad \frac{\mathcal{G} \vdash A : B \quad \mathcal{G} \vdash C : s}{\mathcal{G}, x : C \vdash A : B} \\
\\
(\text{product}) \quad \frac{\mathcal{G} \vdash A : s_1 \quad \mathcal{G}, x : A \vdash B : s_2}{\mathcal{G} \vdash (\Pi x : A.B) : s_3} \quad \text{if } (s_1, s_2, s_3) \in \mathcal{R} \\
\\
(\text{application}) \quad \frac{\mathcal{G} \vdash F : (\Pi x : A.B) \quad \mathcal{G} \vdash a : A}{\mathcal{G} \vdash Fa : B[x := a]} \\
\\
(\text{abstraction}) \quad \frac{\mathcal{G}, x : A \vdash b : B \quad \mathcal{G} \vdash (\Pi x : A.B) : s}{\mathcal{G} \vdash (\lambda x : A.b) : (\Pi x : A.B)} \\
\\
(\text{conversion}) \quad \frac{\mathcal{G} \vdash A : B \quad \mathcal{G} \vdash B' : s \quad B =_\beta B'}{\mathcal{G} \vdash A : B'} \\
\\
(\text{boxing}) \quad \frac{\mathcal{G} \vdash A : \text{Prop}}{\mathcal{G} \vdash \square^\circ A : \text{Prop}} \quad (\text{if } o \in \mathcal{O}) \\
\\
(\text{transfer}_1) \quad \frac{\mathcal{G} \vdash A : s}{\mathcal{G} \square^\circ \varepsilon \vdash A : s} \quad (\text{if } o \in \mathcal{O}) \\
\\
(\text{transfer}_2) \quad \frac{\mathcal{G} \vdash A : B : \text{Type}}{\mathcal{G} \square^\circ \varepsilon \vdash A : B} \quad (\text{if } o \in \mathcal{O}) \\
\\
(\text{transfer}_3) \quad \frac{\mathcal{G} \vdash A : B : \text{Set}}{\mathcal{G} \square^\circ \varepsilon \vdash A : B} \quad (\text{if } o \in \mathcal{O}) \\
\\
(\text{transfer}_{ax}) \quad \frac{\mathcal{G} \vdash c : A : \text{Prop}}{\mathcal{G} \square^\circ \varepsilon \vdash c : A} \quad (\text{if } o \in \mathcal{O} \text{ and } c : A \in \mathcal{A}^{\text{Logic}}) \\
\\
(K \text{ import}) \quad \frac{\mathcal{G} \vdash A : \square^\circ B : \text{Prop}}{\mathcal{G} \square^\circ \varepsilon \vdash \check{k}^\circ A : B} \quad (\text{if } o \in \mathcal{O})
\end{array}$$

In [Borghuis 1994] generalized contexts were denoted by the letter G , here we denote them by \mathcal{G} because of the clash with the established use of G for the forward looking universal tense operator.

$$(K \text{ export}) \quad \frac{\mathcal{G} \boxplus^o \varepsilon \vdash A : B : Prop}{\mathcal{G} \vdash \hat{k}^o A : \square^o B} \quad (\text{if } o \in \mathcal{O})$$

s ranges over the \mathcal{S} the set of sorts, x ranges over variables, c over constants, o ranges over the set \mathcal{O} of modal indices, and it is assumed that in the rules (*start*) and (*weakening*), the newly declared variable is always fresh.

The rules up to *conversion* (with the exception of *axioms* for the ‘logical axioms’ \mathcal{A}^{Logic}) are familiar; they are the PTS-rules stated with respect to generalized contexts (\mathcal{G}) rather than ‘ordinary’ contexts (Γ). The rule *boxing* allows the formation of a ‘modal type’ $\square^o A$ for some operator in \mathcal{O} if this type is a proposition ($A : Prop$). The rest of the rules use the additional structure of the generalized context.

In modal Fitch-style deduction, modal subordinate proofs are used to restrict the reiteration rule to formulas of a certain modal form. In MPTSs this is achieved by means of the separators. In a generalized context $\mathcal{G} \boxplus \Gamma$, we call \mathcal{G} the ‘main context’ and ‘ Γ ’ the subordinate context. The *K-import* rule states that only statements representing proof/proposition pairs for propositions of the form $\square^o B$ ($A : \square^o B : Prop$) in the main context may be repeated in the subordinate context with their type ‘demodalized’ ($B : Prop$). The rule switches the context of derivation from \mathcal{G} to the *empty* subordinate context $\mathcal{G} \boxplus \varepsilon$ to indicate that *K-import* by itself does not require a hypothesis in the subordinate proof (assumptions can be introduced in the subordinate context using *start* and *weakening*). In the *K-export* rule it is essential that the subordinate context is empty: Fitch-style *K-export* requires that the formula to which it is applied has a *categorical* proof in the modal subordinate proof, i.e. the subordinate proof has no undischarged assumptions. Type theoretically this means that a statement $A : B : Prop$ must be derivable on the empty subordinate context $\mathcal{G} \boxplus^o \varepsilon$ before it may be brought back to the main context with its type modalized ($\square^o B$).

Besides the propositions (types) and interval structure (context) of Fitch-style deduction, MPTSs also have terms inhabiting the types. Under the propositions-as-types interpretation, terms represent a proof of the proposition represented by their type. Steps in the proof (like \supset -introduction or elimination) are ‘recorded’ in the structure of the term (as applications and abstractions). To record *K-import* and export steps in modalized deduction, the MPTS-rules change the terms by means of the ‘modal functions’ ‘ \check{k}^o ’ for import and ‘ \hat{k}^o ’ for export. With the Kripke semantics of modal logic in mind, their effect can be described as follows:

- \check{k}^o : import ‘specializes’ a proof A of B for all accessible o -worlds ($\square^o B$) into a proof of B for the arbitrary o -accessible world represented by the subordinate context.
- \hat{k}^o : export ‘generalizes’ a proof A of B in an arbitrary o -world (the subordinate context) into a proof of B for all accessible o -worlds and hence into a proof of $\square^o B$ in the main context.

Hence for statements representing proof/proposition pairs, the transfer between the main and subordinate context is restricted in precisely the same way as the transfer of propositions between the main and modal subordinate proof in Fitch-style deduction. The *transfer* rules 1–3 ensure that this restriction does not apply to the rest of the statements; for these statements the separators in generalized contexts are irrelevant, they behave in the same way as they do in PTSs with respect to ‘ordinary’ contexts.

The MPTS definition above is for multi-modal systems which have the minimal modal logic *K* for all operators in \mathcal{O} . As in Fitch-style deduction, one way to increase the modal

strength is to add further import and export rules. We give the type theoretical versions of the additional rules of section 3.1.

4.4. DEFINITION. Additional modal rules

$$\begin{array}{l}
(4 \text{ import}) \quad \frac{\mathcal{G} \vdash A : \Box^\circ B : Prop}{\mathcal{G} \Box^\circ \varepsilon \vdash \check{4}^\circ A : \Box^\circ B} \quad (\text{if } o \in \mathcal{O}_{4 \text{ import}}(\subseteq \mathcal{O})) \\
(5 \text{ import}) \quad \frac{\mathcal{G} \vdash A : \neg \Box^\circ B : Prop}{\mathcal{G} \Box^\circ \varepsilon \vdash \check{5}^\circ A : \neg \Box^\circ B} \quad (\text{if } o \in \mathcal{O}_{5 \text{ import}}(\subseteq \mathcal{O})) \\
(B \text{ import}) \quad \frac{\mathcal{G} \vdash A : B : Prop}{\mathcal{G} \Box^\circ \varepsilon \vdash \check{b}^\circ A : \neg \Box^\circ \neg B} \quad (\text{if } o \in \mathcal{O}_{B \text{ import}}(\subseteq \mathcal{O})) \\
(D \text{ export}) \quad \frac{\mathcal{G} \Box^\circ \varepsilon \vdash A : B : Prop}{\mathcal{G} \vdash \check{d}^\circ A : \neg \Box^\circ \neg B} \quad (\text{if } o \in \mathcal{O}_{D \text{ export}}(\subseteq \mathcal{O})) \\
(T \text{ export}) \quad \frac{\mathcal{G} \Box^\circ \varepsilon \vdash A : B : Prop}{\mathcal{G} \vdash \check{i}^\circ A : B} \quad (\text{if } o \in \mathcal{O}_{T \text{ export}}(\subseteq \mathcal{O}))
\end{array}$$

Note that each rule introduces its own modal function, requiring the set of pseudoterms to include $\check{4}^\circ T$, $\check{5}^\circ T$, $\check{b}^\circ T$, $\check{d}^\circ T$, and $\check{i}^\circ T$.

Like PTSs, MPTSs are parametrized by sorts (\mathcal{S}), axioms (\mathcal{A}) and rules (\mathcal{R}). The MPTS we use for multi-modal propositional logics is $\lambda \Box PROP2$, the modal version of the PTS $\lambda PROP2$ ([Geuvers 1993]). The latter system is part of the ‘Logic Cube’, a family of PTSs specifically tailored for the interpretation of (non-modal) logics. It has sorts for propositions ($Prop$), sets ($Sets$) and their supertypes ($Type^p$, $Type^s$) related by the *type axioms* $Prop : Type^p$ and $Set : Type^s$. The rules \mathcal{R} of $\lambda PROP2$ and $\lambda \Box PROP2$ are $(Prop, Prop, Prop)$ and $(Type^p, Prop, Prop)$ which substituted for s_1, s_2, s_3 in *product*, *application*, and *abstraction* allows for the formation, elimination and introduction of propositional implication ($A \supset B := \Pi x : A. B : Prop$, for $A, B : Prop$) and universal quantification over propositions ($\forall \alpha \in Prop. B := \Pi \alpha : Prop. B$ for $B : Prop$).

$\lambda PROP2$ corresponds to second order *intuitionistic* propositional logic and is the PTS standardly used for the interpretation of *classical* propositional logic (see [Geuvers 1993]). Since PTSs are inherently intuitionistic, the rule of double negation elimination of classical logic has no counterpart in these systems. Using the quantification over propositional types, the double negation rule becomes expressible as an axiom schema: $c : (\forall \alpha \in Prop. ((\alpha \supset \perp) \supset \perp) \supset \alpha)$, where ‘ \perp ’ is defined as $\forall \alpha \in Prop. \alpha$. Adding this statement to $\lambda \Box PROP2$ as a ‘logical axiom’ ($\in \mathcal{A}^{Logic}$), gives us a modal type system with an underlying classical propositional logic. Unlike in PTSs, we cannot simply treat logical axioms as elements of the initial context of a derivation. In MPTSs, we have to distinguish them from other statements representing proof/proposition pairs ($A : B : Prop$), since for these statements traffic between the main and subordinate contexts is restricted whereas the logical axioms should be available everywhere (which is guaranteed by *transfer_{ax}*).

We will not go into the formal details of the correspondence between modal Fitch-style proofs and MPTS-terms. The following sections provide the reader with enough examples to capture the intuition underlying it. Formalizing this intuition requires a lot of what Girard would call ‘bureaucracy’, and therefore we summarize these formalities by means of two theorems from [Borghuis 1994], relating $\lambda \Box PROP2$ to the Fitch-style system $\Box PROP2$ which is

$\Box PROP_{Fitch}^{\mathcal{O}}$ (section 3.1) with universal quantification over propositions, and ‘ \supset ’ and ‘ \perp ’ as its set of connectives.

THEOREM. *If Σ is a natural deduction proof of φ in $\Box PROP2$, then $\Gamma_{\Sigma} \vdash \Sigma^{\dagger} : \varphi$ is derivable in $\lambda\Box PROP2$ (where Γ_{Σ} is a (non-blocked) context depending on Σ).*

THEOREM. *If $\Gamma \vdash M : \varphi : PROP$ for a term M in $\lambda\Box PROP2$, then $M^?$ is a natural deduction proof of φ in $\Box PROP2$ (where Γ is non-blocked, and all (open) hypotheses of $M^?$ are declared in Γ).*

Hence we have formal mappings from Fitch-style proofs to MPTS-terms (‘ $!$ ’) and from MPTS-terms to Fitch-style proofs (‘ $?$ ’), both of which are sound.

4.2. The MPTS λK_t

In defining an MPTS (λK_t) for the tense logic K_t , we adopt the same strategy as used for the definition of the Fitch-style deduction system for this logic (section 3.2); we start from the minimal multi-modal type system for the two tense operators:

Sorts: $\mathcal{S} = \{Prop, Set, Type^p, Type^s\}$

Axioms:

$\mathcal{A}^{Type} = \{Prop : Type^p, Set : Type^s\}$

$\mathcal{A}^{Logic} = \{c : (\forall \alpha \in Prop. ((\alpha \supset \perp) \supset \perp) \supset \alpha)\}$

Rules: $\mathcal{R} = \{(Prop, Prop, Prop), (Type^p, Prop, Prop)\}$

Operators: $\mathcal{O} = \{G, H\}$.

The triple $(\mathcal{S}, \mathcal{A}, \mathcal{R})$ determines the MPTS $\lambda\Box PROP2^{\mathcal{O}}$, in which the minimal modal logic K can be interpreted for each of the operators in \mathcal{O} . As in the Fitch-style system, having the K -rules for ‘ G ’ and ‘ H ’ suffices to account for the normality axioms and the *Necessitation*-rules. To provide the reader with an opportunity to see the MPTS at work, we construct an inhabitant for the axiom $G(A \supset B) \supset (GA \supset GB)$. (The Start lemma used in the derivation will be explained below.)

All other connectives can be defined using \forall , \supset and \perp .

1. $\Gamma \vdash G(A \supset B) : Prop$ (Start lemma)
2. $\Gamma, x : G(A \supset B) \vdash x : G(A \supset B)$ (start 1)
3. $\Gamma, x : G(A \supset B) \vdash GA : Prop$ (Start lemma)
4. $\Gamma, x : G(A \supset B), y : GA \vdash y : GA$ (start 3)
5. $\Gamma, x : G(A \supset B), y : GA \vdash x : G(A \supset B)$ (Start lemma)
6. $\Gamma, x : G(A \supset B), y : GA \boxtimes^G \varepsilon \vdash \check{k}^G x : A \supset B$ (K -import 5)
7. $\Gamma, x : G(A \supset B), y : GA \boxtimes^G \varepsilon \vdash \check{k}^G y : A$ (K -import 4)
8. $\Gamma, x : G(A \supset B), y : GA \boxtimes^G \varepsilon \vdash \check{k}^G x(\check{k}^G y) : B$ (appl. 6,7)
9. $\Gamma, x : G(A \supset B), y : GA \vdash \hat{k}^G(\check{k}^G x(\check{k}^G y)) : GB$ (K -export 8)
10. $\Gamma, x : G(A \supset B) \vdash \lambda y : GA. \hat{k}^G(\check{k}^G x(\check{k}^G y)) : GA \supset GB$ (abstr. 4,9)
11. $\Gamma \vdash \lambda x : G(A \supset B). (\lambda y : GA. \hat{k}^G(\check{k}^G x(\check{k}^G y))) : G(A \supset B) \supset (GA \supset GB)$ (abstr. 2,10)

If we look just at the types to the right of ‘ \vdash ’, we see that from line 4 on down the type derivation is analogous to the natural deduction proof for $G(A \supset B) \supset (GA \supset GB)$ in section 3.2 (keep in mind that the presence of ‘ \boxtimes^G ’ in the context signifies a G -subordinate proof). Lines 1 and 3 abbreviate the derivation of the well-typedness of $G(A \supset B)$ and GA : we have simply assumed that the context Γ already contains these statements, and we use a derived rule (the Start lemma) that allows us to say that any statement that is an element of a non-blocked context is derivable on that context regardless of its position in it. After the well-formedness of these types has been established, we introduce variables inhabiting them. The variables ‘ x ’ and ‘ y ’ act as ‘dummy proof objects’; adding the statements $x : G(A \supset B)$ and $y : GA$ to the context is the type theoretical analogon of opening hypothesis intervals with hypotheses $G(A \supset B)$ and GA in the natural deduction proof.

However, the fundamental correspondence between the logic and the type theory is that between entire natural deduction proofs in $\square PROP2_{Fitch}^G$ and single terms in $\lambda \square PROP2^G$: the natural deduction proof of $G(A \supset B) \supset (GA \supset GB)$ is represented in the proof object $\lambda x : G(A \supset B). (\lambda y : GA. \hat{k}^G(\check{k}^G x(\check{k}^G y)))$ as it occurs in the final line 11. The idea is that the natural deduction proof can be ‘reconstructed from the bottom up’ by reading the λ -term ‘from the outside in’. The outermost elements of the term are the λ -abstractions over $x : G(A \supset B)$ and $y : GA$. These correspond to applications of \supset -intro in natural deduction, discharging hypotheses $G(A \supset B)$ and (before that) GA . The remaining term $\hat{k}^G(\check{k}^G x(\check{k}^G y))$ codes a proof of GB . By the outermost function \hat{k}^G , the last step in this proof was an application of K -export from a G -subordinate proof. This G -subordinate proof of B is represented by the application term $(\check{k}^G x)(\check{k}^G y)$, hence the last applied rule was \supset -elimination with $\check{k}^G x$ proving $(A \supset B)$ and $\check{k}^G y$ proving A . Outermost in both terms is the function \check{k}^G which shows that they were obtained by an application of K -import into a G -subordinate proof. Hence we are left with two ‘atomic’ proof objects: we cannot decompose the variables $x(: G(A \supset B))$ and $y(: GA)$ any further, they are inhabitants of hypotheses of the natural deduction proof.

Now that the correspondence between the minimal 2-operator modal logic and the minimal 2-operator MPTS has been established, we still have to deal type theoretically with the

The full derivation of $G(A \supset B) : Prop$ and $GA : Prop$ would start from the *type axiom* : $\varepsilon \vdash Prop : Type^P$. By applications of *start* a context $A : Prop, B : Prop$ can be created on which $A \supset B : Prop$ ($\Pi x : A. B : Prop$) is derivable using *product*. Subsequent applications of *Boxing* then yield $G(A \supset B) : Prop$ and $GA : Prop$.

symmetry axioms. We do this in the same way as in Fitch-style deduction, by adding an import rule to λK_t (cf. definition 3.11):

$$(B2\ import) \frac{\mathcal{G} \vdash A : B : Prop}{\mathcal{G} \boxtimes^o \varepsilon \vdash \check{e}^{(o,o')} A : \neg \square^{o'} \neg B} \quad (\text{if } o, o' \in \mathcal{O}_{B2\ import}(\subseteq \mathcal{O} \times \mathcal{O})).$$

Clearly this rule changes the types of statements in the same way as the equinominous Fitch-style rule changes propositions: a type B is transformed into $\neg \square^{o'} \neg B$ in an o -subordinate context (cf. section 3.2). The proof object A is prefixed with the function $\check{e}^{(o,o')}$, which records the modality of the subcontext into which the statement is imported in the index o , and the modal operator that is prefixed to the type in the index o' . If we instantiate the import rule for the set $\mathcal{O}_{B2\ import}$ of K_t ($= \{(G, H), (H, G)\}$) we get the following two rules:

$$\frac{\mathcal{G} \vdash A : B : Prop}{\mathcal{G} \boxtimes^G \varepsilon \vdash \check{e}^{(G,H)} A : \neg H \neg B} \quad \frac{\mathcal{G} \vdash A : B : Prop}{\mathcal{G} \boxtimes^H \varepsilon \vdash \check{e}^{(H,G)} A : \neg G \neg B}.$$

In combination with the appropriate K -export rule, these immediately give us inhabitants for the symmetry axioms: $\lambda x : \varphi. \hat{k}^G(\check{e}^{(G,H)} x) : \varphi \supset G \neg H \neg \varphi$, $\lambda x : \varphi. \hat{k}^H(\check{e}^{(H,G)} x) : \varphi \supset H \neg G \neg \varphi$.

The examples given above do not yet show that the MPTS λK_t corresponds in any precise formal way to the Fitch-style deduction system for K_t , and we cannot conclude it directly from the theorems relating multi-modal Fitch-style deduction systems to multi-modal MPTSs in the previous section, since they are concerned with systems without $B2$ -import. However, the proofs of these results (see [Borghuis 1994]) can easily be redone for $\square PROP2_{Fitch}^{\mathcal{O}}$ and $\lambda \square PROP2^{\mathcal{O}}$ extended with $B2$ -import.

4.3. Further properties of the tenses

For the extension of λK_t with further properties of the tenses, we have the two options discussed for Fitch-style deduction in 3.3:

Extension by axioms

Tense logical principles can be added to λK_t as logical axioms ($\in \mathcal{A}^{Logic}$). The possibility of quantifying over propositional variables allows us to use axiom schemas rather, e.g. $c_{4_p} : (\forall \alpha \in Prop(H\alpha \supset HH\alpha))$. From such a schema, the axiom for a particular type (say $A : Prop$) can be obtained by applying the schema to that type ($c_{4_p} A : HA \supset HHA$). The resulting axiom can then be used as in natural deduction proofs; by applying it to a proof of the antecedent of the axiom ($M : HA$) a proof of the consequent is obtained ($((c_4 A)M : HHA)$). Logical axioms may be repeated anywhere in a derivation because our rules interact to ensure that they are derivable on any well-formed context: by *axiom* logical axioms are derivable on the empty context ε , hence they are derivable on any non-blocked context Γ by (repeated) *weakening*, and by *transfer_{ax}* they can be lifted over \boxtimes 's into subordinate contexts.

Extension by rules

Like in Fitch-style deduction, adding import and export rules to λK_t strengthens the system by allowing for more ways of transporting statements from the main to the subordinate context and back. The Fitch-style rules indicate how the types of statements behave under import or export, the terms change in such a way that they record all information needed to reconstruct the modal step in the natural deduction proof (like the modality of the imported

formula and that of the subordinate proof).

In section 4.1 a number of additional modal rules are given which are the type theoretical counterpart of the mono-modal Fitch-style rules of section 3.1. Hence we can use these to deal straightforwardly with the tense logical properties $4_f/4_p$, T_f/T_p and D_f/D_p of section 2.2.1. We show this in detail for 4_p ; given λK_t and the 4-import rule for H , an inhabitant for $H\varphi \supset HH\varphi$ can be constructed.

$$(4_H\text{-import}) \quad \frac{\mathcal{G} \vdash A : HB : Prop}{\mathcal{G} \boxtimes^H \varepsilon \vdash \check{4}^H A : HB}$$

1. $\varepsilon \vdash Prop : Type^p$ (axiom)
2. $\varphi : Prop \vdash \varphi : Prop$ (start 1)
3. $\varphi : Prop \vdash H\varphi : Prop$ (boxing 2)
4. $\varphi : Prop, x : H\varphi \vdash x : H\varphi$ (start 3)
5. $\varphi : Prop, x : H\varphi \boxtimes^H \varepsilon \vdash \check{4}^H x : H\varphi$ (4-import 4)
6. $\varphi : Prop, x : H\varphi \vdash \hat{k}^H(\check{4}^H x) : HH\varphi$ (K -export 5)
7. $\varphi : Prop \vdash \lambda x : H\varphi. \hat{k}^H(\check{4}^H x) : H\varphi \supset HH\varphi$ (abstr. 6)

The corresponding proof for 4_f is left to the reader.

Similarly by combining T -export (for G) and D -export (for H) with K -import, we obtain proof objects for $T_f: G\varphi \supset \varphi$ and $D_p: H\varphi \supset \neg H\neg\varphi$

$$(T\text{-export}) \quad \frac{\mathcal{G} \boxtimes^G \varepsilon \vdash A : B : Prop}{\mathcal{G} \vdash \hat{i}^G A : B} \quad \lambda x : G\varphi. \hat{i}^G(\check{k}^G x) : G\varphi \supset \varphi$$

$$(D\text{-export}) \quad \frac{\mathcal{G} \boxtimes^H \varepsilon \vdash A : B : Prop}{\mathcal{G} \vdash \hat{d}^H A : \neg H\neg B} \quad \lambda x : H\varphi. \hat{d}^H(\check{k}^H x) : H\varphi \supset \neg H\neg\varphi$$

The derivation leading to proof objects for T_p and D_f are, again, left to the reader.

For the density axioms, $Dens_f: GG\varphi \supset G\varphi$ and $Dens_p: HH\varphi \supset H\varphi$, we adopt the type theoretical analogon of the Fitch-style rule $Dense$ -export:

$$(Dense\text{-export}) \quad \frac{\mathcal{G} \boxtimes^o \varepsilon \vdash A : \square^o B : Prop}{\mathcal{G} \vdash \hat{v}^o A : \square^o B} \quad \forall o \in O_{Dense\text{-export}} (\subseteq \mathcal{O}) .$$

In this rule \hat{v}^o records that $Dense$ -export was applied to a proof of $\square^o B$ (for some proposition type B) in an o -subordinate proof. In the above format, the rule summarizes $Dense$ -export for both directions of time: instantiating it with G and H gives us two export rules which immediately yield inhabitants for the density axioms.

$$\frac{\mathcal{G} \boxtimes^G \varepsilon \vdash A : GB : Prop}{\mathcal{G} \vdash \hat{v}^G A : GB} \quad \lambda x : GG\varphi. \hat{v}^G(\check{k}^G x) : GG\varphi \supset G\varphi$$

$$\frac{\mathcal{G} \boxtimes^H \varepsilon \vdash A : HB : Prop}{\mathcal{G} \vdash \hat{v}^H A : HB} \quad \lambda x : HH\varphi. \hat{v}^H(\check{k}^H x) : HH\varphi \supset H\varphi .$$

Since no additional import- or export-rules of the above kind can be given for the remaining principles Z_f/Z_p , W_f/W_p , Dum_f/Dum_p , and $Cont$, we act as in section 3.3 and add them as logical axioms for the time being (this matter will be discussed further in section 5.2):

$$\begin{aligned}
c_{zf} &: (\forall \alpha \in Prop. G(G\alpha \supset \alpha) \supset (\neg G \neg G\alpha \supset G\alpha)) \\
c_{zp} &: (\forall \alpha \in Prop. H(H\alpha \supset \alpha) \supset (\neg H \neg H\alpha \supset H\alpha)) \\
c_{wf} &: (\forall \alpha \in Prop. G(G\alpha \supset \alpha) \supset G\alpha) \\
c_{wp} &: (\forall \alpha \in Prop. H(H\alpha \supset \alpha) \supset H\alpha) \\
c_{Dumf} &: (\forall \alpha \in Prop. G(G(\alpha \supset G\alpha) \supset \alpha) \supset (\neg G \neg G\alpha \supset \alpha)) \\
c_{Dumf} &: (\forall \alpha \in Prop. H(H(\alpha \supset H\alpha) \supset \alpha) \supset (\neg H \neg H\alpha \supset \alpha)) \\
c_{Cont} &: (\forall \alpha \in Prop. \blacksquare(H\alpha \supset \neg G \neg H\alpha) \supset (H\alpha \supset G\alpha))
\end{aligned}$$

In cases for which both an extension by axioms and an extension by rules exists, these two extensions of λK_t are equivalent in the sense that a term inhabiting the axiom schema can be derived in λK_t extended with the rule and a term corresponding to the term introduced by the import- or export-rule is derivable in λK_t extended with the axiom schema. We show this for the principle $4_p : H\varphi \supset HH\varphi$. For a formal translation between terms in the rule-extended system and terms in the axiom-extended system the reader is referred to [Borghuis 1994].

From rule to axiom schema:

In λK_t extended with 4_H -import, an inhabitant for the type of the logical axiom $c_{4p} : (\forall \alpha \in Prop.(H\alpha \supset HH\alpha))$ can be derived on the empty context, as can be seen as follows: the example derivation for the 4-axiom above ended in $\varphi : Prop \vdash \lambda x : H\varphi. \hat{k}^H(\check{4}^H x) : H\varphi \supset HH\varphi$. It can be continued by abstracting over the propositional variable φ , resulting in $\varepsilon \vdash \lambda \varphi : Prop. \lambda x : H\varphi. \hat{k}^H(\check{4}^H x) : (\forall \varphi \in Prop. H\varphi \supset HH\varphi)$ in which the term represents a proof of the axiom schema (modulo α -equivalence).

From axiom schema to rule:

In λK_t extended with the logical axiom $c_{4p} : (\forall \alpha \in Prop.(H\alpha \supset HH\alpha))$ the 4_H -import rule can be ‘mimicked’; starting from a statement $M : H\varphi$ in some context G we can obtain a statement of type $H\varphi$ in the subordinate context which has M as a subterm, using K -import for H as the only modal rule:

1. $\mathcal{G} \vdash M : H\varphi$ (assumption)
2. $\mathcal{G} \vdash c_{4p} : (\forall \alpha \in Prop.(H\alpha \supset HH\alpha))$ (4_p -axiom)
3. $\mathcal{G} \vdash \varphi : Prop$ Start lemma
4. $\mathcal{G} \vdash (c_{4p} \varphi) : H\varphi \supset HH\varphi$ (appl. 2,3)
5. $\mathcal{G} \vdash (c_{4p} \varphi)M : HH\varphi$ (appl. 1,4)
6. $\mathcal{G} \boxplus^H \varepsilon \vdash \check{k}^H((c_{4p} \varphi)M) : H\varphi$ (K -import 5)

4.4. The interaction between past and future

In the logic K_t , the basic interaction between the past and future directions of time is given by the symmetry between G and H . A Fitch-style analysis of the deductive effects of this symmetry in systems for K_t and extensions was given in section 3.4. Since MPTSs correspond closely to the Fitch-style deduction systems, this analysis carries over to λK_t . Hence we cannot expect new insights into the effect of symmetry on the level of types. However, on the level of terms some aspects of the Fitch-style analysis can be made more precise.

In 3.4.1 an informal explanation of the mirror image property for the deduction system for K_t was given: from a proof of $\varphi(G, H)$ we can construct a proof of $\varphi(H, G)$ by taking the mirror image of all hypotheses occurring in the original proof and then matching all rule applications in the original proof step by step with applications of the mirror images of these rules in the new proof. Since in λK_t terms represent an entire natural deduction proof of their proposition type, this operation can be viewed as a substitution on terms which given a proof object M of type $\varphi(G, H)$ yields a proof object M' inhabiting $\varphi(H, G)$.

PROPOSITION. If the λK_t -term M corresponds to a natural deduction proof (Σ) of $\varphi(G, H)$ in K_t ($\Gamma_\Sigma \vdash M : \varphi(G, H)$), a λK_t -term M' corresponding to a natural deduction proof of $\varphi(H, G)$ in K_t ($\Gamma'_\Sigma \vdash M' : \varphi(H, G)$) can be obtained by simultaneously replacing all occurrences in M (and Γ_Σ) of:

- (i) G by occurrences of H ,
 H by occurrences of G ,
- (ii) \hat{k}^G by occurrences of \hat{k}^H / \check{k}^H by occurrences of \hat{k}^G ,
 \check{k}^G by occurrences of \check{k}^H / \hat{k}^H by occurrences of \check{k}^G ,
 $\tilde{e}^{(G,H)}$ by occurrences of $\tilde{e}^{(H,G)}$ / $\tilde{e}^{(H,G)}$ by occurrences of $\tilde{e}^{(G,H)}$.

Clause (i) substitutes the mirror images for the hypotheses of the natural deduction proof Σ , which occur in the proof object M as the type of a bound variable (if the hypothesis is discharged in Σ) or as the type of a declaration in the context Γ_Σ (if the hypothesis is not discharged). The second clause replaces every modal function in M representing an application of an import or export rule in Σ by the modal function representing an application of the mirror image of this import or export rule.

An example of a pair of proof objects that are each others mirror image under this translation are the inhabitants of the normality axioms:

$$\lambda x : G(A \supset B). \lambda y : GA. \hat{k}^G((\check{k}^G x)(\check{k}^G y)) : G(A \supset B) \supset (GA \supset GB)$$

$$\lambda x : H(A \supset B). \lambda y : HA. \hat{k}^H((\check{k}^H x)(\check{k}^H y)) : H(A \supset B) \supset (HA \supset HB) .$$

The effects of symmetry on the extensions of K_t highlighted by the examples 1 and 2 of 3.4.1 are equally present in λK_t :

In λK_t plus 4-import for G an inhabitant of $H\varphi \supset HH\varphi$ can be constructed.

In λK_t plus *Dens*-export for H an inhabitant of $GG\varphi \supset G\varphi$ can be constructed.

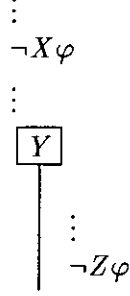
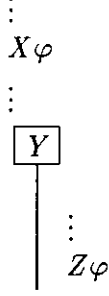
We leave it as an exercise for the industrious reader to actually derive the proof objects corresponding to the Fitch-style proofs in Examples 1 and 2.

For a more general view on the interaction between the past and future direction of time, we return to the classification of interaction axioms in categories positive/negative intro/extraspection. In section 3.4 it was shown how Fitch-style rules can be given for each of these categories, and by now the reader will not be surprised that we can match these with MPTS-rules. Below we give an overview for introspection and extraspection, showing from top to bottom: the interactions axiom, the Fitch-style rule, the MPTS-rule, and an inhabitant of the axiom derived by means of the latter rule.

Introspection

$$X\varphi \supset YZ\varphi$$

$$\neg X\varphi \supset Y\neg Z\varphi$$



positive introspection negative introspection

$$\text{pos introspection} \frac{\mathcal{G} \vdash M : X\varphi : Prop}{\mathcal{G} \boxtimes^Y \varepsilon \vdash \check{p}^{(X,Y,Z)} M : Z\varphi}$$

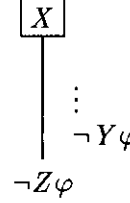
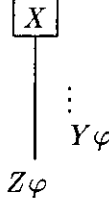
$$\text{neg introspection} \frac{\mathcal{G} \vdash M : \neg X\varphi : Prop}{\mathcal{G} \boxtimes^Y \varepsilon \vdash \check{n}^{(X,Y,Z)} M : \neg Z\varphi}$$

$$\lambda u : X\varphi. \hat{k}^Y(\check{p}^{(X,Y,Z)} u) : X\varphi \supset YZ\varphi \quad \lambda u : \neg X\varphi. \hat{k}^Y(\check{n}^{(X,Y,Z)} u) : \neg X\varphi \supset Y\neg Z\varphi.$$

Extraspection

$$XY\varphi \supset Z\varphi$$

$$X\neg Y\varphi \supset \neg Z\varphi$$



positive extraspection negative extraspection

$$\text{pos extraspection} \frac{\mathcal{G} \boxtimes^X \varepsilon \vdash M : Y\varphi : Prop}{\mathcal{G} \vdash \hat{p}^{(X,Y,Z)} M : Z\varphi}$$

$$\text{neg extraspection} \frac{\mathcal{G} \boxtimes^X \varepsilon \vdash M : \neg Y\varphi : Prop}{\mathcal{G} \vdash \hat{n}^{(X,Y,Z)} M : \neg Z\varphi}$$

$$\lambda u : XY\varphi. \hat{p}^{(X,Y,Z)}(\check{k}^X u) : XY\varphi \supset Z\varphi \quad \lambda u : X\neg Y\varphi. \hat{n}^{(X,Y,Z)}(\check{k}^X u) : X\neg Y\varphi \supset \neg Z\varphi.$$

Note that the proof functions $(\check{p}, \hat{p}, \check{n}, \hat{n})$ in these MPTS-rules are indexed with triples of operator indices to record all relevant aspects of the application of the Fitch-style rules: the modality of the formula to which the rule is applied, the modality of the strict subordinate proof and the modality of the resulting formula. As for the Fitch-style rules, all modal MPTS-rules defined sofar fit in the classification by identifying or leaving out combinations of operators (in the types) and indices X, Y, Z (in the terms), and replacing n and p by more mnemonic letters.

In section 3.4.2, we discussed two cases where an extension of K_t with an interaction axiom of one category made an axiom of a different category derivable due to the symmetry between G and H . Naturally these examples go through for λK_t :

- in $\lambda K_t + 4$ -import inhabitants for the preservation axioms, $\neg G\neg\varphi \supset H\neg G\neg\varphi$ and $\neg H\neg\varphi \supset G\neg H\neg\varphi$ can be constructed.
- in $\lambda K_t + 4$ -import + D -export inhabitants for the actuality axioms, $GH\varphi \supset H\varphi$ and $HG\varphi \supset G\varphi$ can be constructed.

The details of this are left to the industrious reader who should be able to derive these proof objects on the basis of the Fitch-style proofs in 3.4.2.

4.5. Linear MPTSs

Given the type theoretical version of the classification of interaction axioms, we can simply follow the Fitch-style analysis of section 3.5 in defining MPTSs corresponding to the minimal linear tense logic Lin and its extensions.

4.5.1. The MPTS λLin

The first step in extending λK_t to λLin , the MPTS corresponding to the minimal linear tense logic, is to include the 4-import rule for G and H , as the ordering of time points is transitive in linear logics ($4_p : H\varphi \supset HH\varphi$, $4_f : G\varphi \supset GG\varphi$).

The second step is to add the universal modality, ‘ \blacksquare ’, to the set of operators, $\mathcal{O} = \{G, H, \blacksquare\}$, in order to deal with the linearity axioms $L_p : \blacksquare\varphi \supset GH\varphi$, and $L_f : \blacksquare\varphi \supset HG\varphi$. Adding ‘ \blacksquare ’ to \mathcal{O} automatically gives us K -import and K -export for this operator, which supplemented with 4-import and T -export suffices to account for its ‘pure’ properties:

$$\begin{aligned} \lambda x : \blacksquare(A \supset B). \lambda y : \blacksquare A. \hat{k}^\blacksquare((\check{k}^\blacksquare x)(\check{k}^\blacksquare y)) : \blacksquare(A \supset B) \supset (\blacksquare A \supset \blacksquare B) \\ \lambda x : \blacksquare A. \hat{k}^\blacksquare(\check{4}^\blacksquare x) : \blacksquare A \supset \blacksquare\blacksquare A \\ \lambda x : \blacksquare A. \hat{t}^\blacksquare(\check{k}^\blacksquare x) : \blacksquare A \supset A . \end{aligned}$$

For the ‘mixed’ principles relating ‘ \blacksquare ’ to ‘ G ’ and ‘ H ’, we need the type theoretical analogon of the positive introspection rules $K2$ -import and U -import proposed in 3.5.1:

$$\begin{array}{l} K2\text{-import} \quad \frac{\mathcal{G} \vdash A : \square^\circ B : Prop}{\mathcal{G} \boxtimes^{o'} \varepsilon \vdash \check{f}^{(o,o')} A : B} \quad \text{if } o, o' \in \mathcal{O}_{K2\text{-import}} \\ U\text{-import} \quad \frac{\mathcal{G} \vdash A : \square^\circ B : Prop}{\mathcal{G} \boxtimes^{o'} \varepsilon \vdash \check{u}^{(o,o',o'')} A : \square^{o''} B} \quad \text{if } o, o', o'' \in \mathcal{O}_{U\text{-import}} . \end{array}$$

The modal function $\check{f}^{(o,o')}$ introduced by $K2$ -import records the modal operator of the imported type in the index o , and the modality of the subordinate context in the index o' . The function $\check{u}^{(o,o',o'')}$ introduced by U -import does the same and in addition stores the operator prefixed to the imported type in the index o'' . If we instantiate these rules according to the sets $\mathcal{O}_{K2\text{-import}} (= \{(\blacksquare, G), (\blacksquare, H)\})$ and $\mathcal{O}_{U\text{-import}} (= \{(\blacksquare, G, H), (\blacksquare, H, G)\})$ of the Lin , we get two pairs of import rules which immediately give us inhabitants for the mixed axioms of ‘ \blacksquare ’:

As in the natural deduction system, the operator ‘ \blacksquare ’ is definable in λK_t ($\blacksquare\varphi =_{def} \forall \alpha \in Prop. (H\varphi \supset (\varphi \supset (G\varphi \supset \alpha))) \supset \alpha$), but only adding it explicitly yields satisfactory deduction rules for the linearity axioms. Cf. section 4.2.

$$\begin{array}{c}
\text{K2-import} \quad \frac{\mathcal{G} \vdash A : \blacksquare B : Prop}{\mathcal{G} \boxtimes^G \varepsilon \vdash \check{f}(\blacksquare, G) A : B} \qquad \frac{\mathcal{G} \vdash A : \blacksquare B : Prop}{\mathcal{G} \boxtimes^H \varepsilon \vdash \check{f}(\blacksquare, H) A : B} \\
\lambda x : \blacksquare \varphi. \hat{k}^G \check{f}(\blacksquare, G) x : \blacksquare \varphi \supset G \varphi \qquad \lambda x : \blacksquare \varphi. \hat{k}^H \check{f}(\blacksquare, H) x : \blacksquare \varphi \supset H \varphi \\
\text{U-import} \quad \frac{\mathcal{G} \vdash A : \blacksquare B : Prop}{\mathcal{G} \boxtimes^G \varepsilon \vdash \check{u}(\blacksquare, G, H) A : HB} \qquad \frac{\mathcal{G} \vdash A : \blacksquare B : Prop}{\mathcal{G} \boxtimes^H \varepsilon \vdash \check{u}(\blacksquare, H, G) A : GB} \\
\lambda x : \blacksquare \varphi. \hat{k}^G \check{u}(\blacksquare, G, H) x : \blacksquare \varphi \supset GH \varphi \qquad \lambda x : \blacksquare \varphi. \hat{k}^H \check{u}(\blacksquare, H, G) x : \blacksquare \varphi \supset HG \varphi .
\end{array}$$

By adding these two rules, we have covered every way of eliminating the \blacksquare -operator. For its introduction we will have to add a logical axiom, like we did for the Fitch-style system in section 3.5.1. In short, the MPTS λLin corresponding to the minimal linear tense logic can be described as $\lambda \square PROP2^{\mathcal{O}}$, where:

$$\begin{aligned}
\mathcal{O} &= \{(G, H, \blacksquare)\} \\
\mathcal{O}_{B2\text{-import}} &= \{(G, H), (H, G)\} \\
\mathcal{O}_{4\text{-import}} &= \{G, H, \blacksquare\} \\
\mathcal{O}_{K2\text{-import}} &= \{(\blacksquare, G), (\blacksquare, H)\} \\
\mathcal{O}_{U\text{-import}} &= \{(\blacksquare, G, H), (\blacksquare, H, G)\} \\
\mathcal{O}_{T\text{-export}} &= \{\blacksquare\} \\
A^{Logic} &= \{c_{\neg} : (\forall \alpha \in Prop. ((\alpha \supset \perp) \supset \perp) \supset \alpha), c_{\blacksquare} : (\forall \alpha \in Prop. ((H\alpha \supset (\alpha \supset G\alpha)) \supset \blacksquare \alpha))\}
\end{aligned}$$

4.5.2. Familiar linear structures

For each of the linear structures presented in section 2.3.2, we can give an MPTS by extending λLin with the same combinations of additional rules and additional axioms that were used in the Fitch-style systems of section 3.5.2.

- $(\mathbb{Z}, <)$, the MPTS λLin and

additional rules: D -export for G
 D -export for H

logical axioms: $c_{zf} : (\forall \alpha \in Prop. G(G\alpha \supset \alpha) \supset (\neg G \neg G\alpha \supset \alpha))$
 $c_{zp} : (\forall \alpha \in Prop. H(H\alpha \supset \alpha) \supset (\neg H \neg H\alpha \supset H\alpha))$

- $(\mathbb{N}, <)$, the MPTS λLin and

additional rules: D -export for G

logical axioms: $c_{zf} : (\forall \alpha \in Prop. G(G\alpha \supset \alpha) \supset (\neg G \neg G\alpha \supset \alpha))$
 $c_{wp} : (\forall \alpha \in Prop. H(H\alpha \supset \alpha) \supset H\alpha)$

- $(\mathbb{Q}, <)$, the MPTS λLin and

additional rules: D -export for G
 D -export for H
 $Dense$ -export for G
 $Dense$ -export for H

- (Q, \leq) , the MPTS λLin and

additional rules: T -export for G
 T -export for H

- $(R, <)$, the MPTS λLin and

additional rules: D -export for G
 D -export for H
 $Dense$ -export for G
 $Dense$ -export for H

logical axioms: $c_{cont} : (\forall \alpha \in Prop.(H\alpha \supset \neg G\neg H\alpha) \supset (H\alpha \supset G\alpha))$.

4.6. Subject reduction

The fact that MPTS-terms correspond to entire natural deduction proofs implies that operations on these terms *inside* an MPTS correspond to *meta-operations* on natural deduction proofs. More in particular, simplification operations on natural deduction proofs can be expressed type theoretically as reduction rules on terms. For standard typed λ -calculi the most important of these ‘subject reductions’ is β -reduction: β -reduction on a term representing a (Prawitz-style) natural deduction proof corresponds to cut-elimination in that proof. Hence the β -normal form of the term will represent a cut-free proof of the proposition represented by the type of the term.

For MPTSs, the question arises whether simplifications of modal Fitch-style proofs exist which can be specified as subject reduction rules inside the type system. In [Borghuis 1994] a number of combinations of import- and export steps that can cause ‘detours’ in Fitch-style proofs have been identified. These detours show up in the MPTS-terms as patterns of import- and export functions, on which subject reduction rules can be defined. Besides the detours familiar from general modal logic, the tense logics discussed in this paper also contain new ones.

4.6.1. Reduction in K_t

Because of the symmetry between K -import and K -export the application of the import rule on a proposition immediately followed by an application of the export rule does not have any observable effect on that proposition. The proposition has not been used to derive anything in the subordinate proof (no rules have been applied to it between import and export) and all steps in the proof that could have been taken before this ‘detour’ can be taken after it.

Prawitz-style (or ‘tree-form’) natural deduction rules have an explicit cut-rule, in Fitch-style deduction the situation is more complicated, but β -reduction also simplifies the proof (see [Borghuis 1994]).

$$\begin{array}{ccc}
\vdots & & \vdots \\
GA & \Rightarrow & GA \\
\boxed{G} & & \vdots \\
\mid & & \\
A & & \\
\mid & & \\
GA & & \\
\vdots & & \\
\hat{k}^G(\check{k}^G M) & & M
\end{array}$$

Type theoretically there is a difference between the occurrences of GA before and after the detour. If the original proof object for GA is M , then the inhabitant of GA after the detour will be $\hat{k}^G(\check{k}^G M)$. In this term it is recorded that the original proof (M) of the proposition (GA) in the main context which has first been specialized to a proof ($\check{k}^G M$) of the proposition (A) in the G -subordinate context by means of the function \check{k}^G and then generalized back into a proof of the original proposition (GA) in the main context by \hat{k}^G .

Given this signature of a detour, we can define a type theoretical reduction rule to formalize the idea that a combination of subsequent K -import and K -export for a given normal modal operator is pointless in a natural deduction proof.

4.5. DEFINITION. $\hat{k}\check{k}$ reduction : $\hat{k}^o(\check{k}^o M) \Rightarrow M \quad \forall o \in \mathcal{O}$

Combined with the mappings to and from the natural deduction proofs, $\hat{k}\check{k}$ -reduction allows us to eliminate detours in a natural deduction proof in the way depicted above: any sequence of K -import and immediate K -export of a formula can be eliminated from the proof.

In view of the symmetry of the basic modal rules, it is not surprising that we can make a similar observation about sequences in the ‘reverse order’: K -export followed by K -import. Given an occurrence of A in a strict subordinate proof, subsequent applications of K -export and K -import again yield an occurrence of A in a strict subordinate proof.

$$\begin{array}{ccc}
\boxed{G} & & \boxed{G} \\
\mid & \Rightarrow & \mid \\
\vdots & & \vdots \\
A & & A \\
\mid & & \mid \\
GA & & \vdots \\
\boxed{G} & & \\
\mid & & \\
A & & \\
\mid & & \\
\vdots & & \\
\check{k}^G(\hat{k}^G M) & & M
\end{array}$$

Eliminating this detour does not make a difference for the rest of the natural deduction proof; since K -export could be applied to A , we know that the first occurrence of A does not depend on any hypotheses of the modal subordinate proof.

Supposing that the original inhabitant of A is M , the type theoretical signature of such a detour is $\check{k}^G(\hat{k}^G M)$. Hence we can define the following reduction for its elimination.

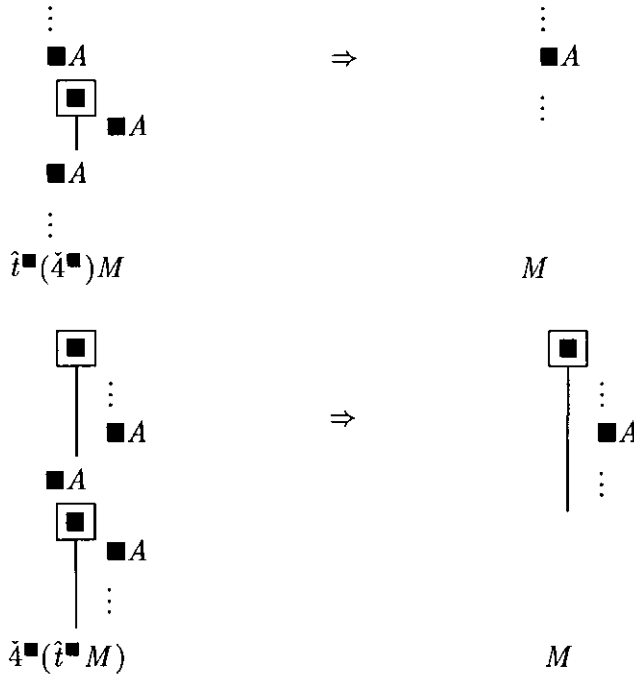
4.6. DEFINITION. $\check{k}\hat{k}$ reduction : $\check{k}^o(\hat{k}^o M) \Rightarrow M \quad \forall o \in \mathcal{O}$

We shall call both kinds of reduction ‘annihilation’; any time a \check{k}^o -function meets a \hat{k}^o -function in any order in a term they ‘destroy’ each other. These reductions are ‘compatible’,

which means that a subterm of the right form (e.g. $\hat{k}^H(\check{k}^H M)$) may always be replaced (by M), regardless of the structure of the term in which it appears (for instance, an application $N(\hat{k}^H(\check{k}^H M))$).

4.6.2. Reduction in *Lin*

In extensions by rules for logics above K_t , new pointless combinations of import and export may arise. In the logic *Lin* this happens for the universal operator ‘■’; for this operator we have 4-import, allowing us to transfer formulas of the form ‘■ φ ’ unchanged to a ■-subordinate proof, and *T*-export, allowing unchanged export of any formula out of a ■-subordinate proof. Immediate subsequent use of these rules leads to detours in the natural deduction proofs of *Lin*:



In *Lin*, these detours can be identified as subterms of the form $\hat{i}^{\blacksquare}(\check{4}^{\blacksquare} M)$ or $\check{4}^{\blacksquare}(\hat{i}^{\blacksquare} M)$ (where M is of type ■ φ). Hence we can formalize their elimination by means of the following subject reduction rules:

4.7. DEFINITION. \hat{i}^{\blacksquare} reduction and $\check{4}^{\blacksquare}$ reduction.

\hat{i}^{\blacksquare} reduction : $\hat{i}^{\blacksquare}(\check{4}^{\blacksquare} M) \Rightarrow M \quad \forall o \in \mathcal{O}$ such that $o \in \mathcal{O}_{T\text{-export}}$ and $o \in \mathcal{O}_{4\text{-import}}$

$\check{4}^{\blacksquare}$ reduction : $\check{4}^{\blacksquare}(\hat{i}^{\blacksquare} M) \Rightarrow M \quad \forall o \in \mathcal{O}$ such that $o \in \mathcal{O}_{4\text{-import}}$ and $o \in \mathcal{O}_{T\text{-export}}$

If the logic *Lin* is strengthened further to accommodate dense flows of time, the above detours can also occur with the weaker operators G and H : in *Lin* we already have 4-import for these operators and in extensions for dense logics *Dens*-export is added. Since *Dens*-export behaves exactly like *T*-export for formulas of the form $G\varphi$ and $H\varphi$, subsequent application of these rules yields the above detours. In the MPTS, these detours can be recognized by the presence of the ‘ $\hat{4}$ ’ import function and ‘ \hat{v} ’ export function.

See [Barendregt 1992].

4.8. DEFINITION. $\hat{v}\check{A}$ reduction and $\check{A}\hat{v}$ reduction.

$\hat{v}\check{A}$ reduction : $\hat{v}^\circ(\check{A}^\circ M) \Rightarrow M \quad \forall o \in \mathcal{O}$ such that $o \in \mathcal{O}_{Dense-export}$ and $o \in \mathcal{O}_{4-import}$

$\check{A}\hat{v}$ reduction : $\check{A}^\circ(\hat{v}^\circ M) \Rightarrow M \quad \forall o \in \mathcal{O}$ such that $o \in \mathcal{O}_{4-import}$ and $o \in \mathcal{O}_{Dense-export}$

Annihilation rules are well-behaved; the combined reductions in an MPTS with β -reduction and annihilations have the same desirable properties as the β -reductions in the original MPTS.

Subject Reduction If $\mathcal{G} \vdash M : A$ and M reduces to M' through a number of annihilations (and β -steps), then $\mathcal{G} \vdash M' : A$: the reduced proof is again a proof of the original formula.

Strong Normalisation For every term M , there is an upperbound to the reductions starting from it: the annihilation reductions of proofs terminate.

Church Rosser If a term M and reduces to different terms M' and M'' , then M' and M'' have a common reduct: different reduction paths will eventually lead to the same result.

This was proved for $\hat{k}\check{k}/\check{k}\hat{k}$ -reduction and $\hat{i}\check{A}/\check{A}\hat{i}$ -reduction in [Borghuis 1994], these proofs are easily adapted to include $\hat{v}\check{A}/\check{A}\hat{v}$ -reduction.

Annihilations eliminate pointless combinations of import and export steps, they remove simple ‘local’ detours from modal Fitch-style proofs. For modalities for which T -export holds, more interesting operations on proofs are possible: since T -export does not change the form of the propositions to which it is applied, it can sometimes be interchanged with propositional steps in the proof. An example of this in *Lin*, where we have T -export for ‘ \blacksquare ’, is the exchange of \supset -elim and T -export:

$$\begin{array}{c}
 \vdots \\
 \blacksquare(\blacksquare A \supset \blacksquare B) \\
 \blacksquare A \\
 \boxed{\blacksquare} \\
 \blacksquare A \supset \blacksquare B \quad (K\text{-import}) \\
 \blacksquare A \quad (4\text{-import}) \\
 \blacksquare B \quad (\supset\text{-elim}) \\
 \blacksquare B \quad (T\text{-export}) \\
 \vdots \\
 \hat{i}^\blacksquare((\check{k}^\blacksquare M)(\check{A}^\blacksquare N))
 \end{array}
 \Rightarrow
 \begin{array}{c}
 \vdots \\
 \blacksquare(\blacksquare A \supset \blacksquare B) \\
 \blacksquare A \\
 \boxed{\blacksquare} \\
 \blacksquare A \supset \blacksquare B \quad (K\text{-import}) \\
 \blacksquare A \quad (4\text{-import}) \\
 \blacksquare A \supset \blacksquare B \quad (T\text{-export}) \\
 \blacksquare A \quad (T\text{-export}) \\
 \blacksquare B \quad (\supset\text{-elim}) \\
 \vdots \\
 (\hat{i}^\blacksquare(\check{k}^\blacksquare M))(\hat{i}^\blacksquare(\check{A}^\blacksquare N))
 \end{array}$$

In the proof on the left, two formulas are imported into the \blacksquare -subordinate proof, \supset -elimination is performed and the result is T -exported to the main proof. In the proof on the right, two formulas are imported, immediately T -exported and \supset -elimination is then performed on the results of the export in the main proof. Hence we have shortened the \blacksquare -subordinate proof by permuting the application of \supset -elim and T -export. Intuitively this operation is justified by the reflexivity of the ‘ \blacksquare ’. Since it quantifies over all time points, \blacksquare -subordinate proofs correspond to arbitrary time points and propositional steps inside such

proof involve formulas which are true at all time points. Hence these proof steps could also be carried out at the present time point (in the main proof).

In the MPTS-terms corresponding to the two proofs, applications of T -export are recorded as occurrences of ' \hat{t} '. Hence the permutation of T -export and \supset -elim in the natural deduction proof shows up as a distribution of \hat{t} over application in the terms: $\hat{t}^\blacksquare((\check{k}^\blacksquare M)(\check{4}^\blacksquare N)) \Rightarrow (\hat{t}^\blacksquare(\check{k}^\blacksquare M))(\hat{t}^\blacksquare(\check{4}^\blacksquare N))$, assuming that $M : \blacksquare(\blacksquare A \supset \blacksquare B)$ and $N : \blacksquare A$. At first glance, it may seem that the deduction proof resulting from this reduction is not simpler than the original: its subordinate proof may be shorter but the main proof has more steps than the original. However, a second look reveals a possibility for further simplification in the new proof; it is now obvious that the 4-import and T -export of $\blacksquare A$ is superfluous, we could immediately have used the topmost occurrence of $\blacksquare A$ in the \supset -elimination. This possibility is reflected in the proof term which contains the \hat{t} -redex $\hat{t}^\blacksquare(\check{4}^\blacksquare N)$. Reducing it in the way described above yields a proof that is simpler than the original proof:

$$\begin{array}{ccc}
\begin{array}{c}
\vdots \\
\blacksquare(\blacksquare A \supset \blacksquare B) \\
\blacksquare A \\
\boxed{\blacksquare} \\
\quad \blacksquare A \supset \blacksquare B \quad (K\text{-import}) \\
\quad \blacksquare A \quad (4\text{-import}) \\
\blacksquare A \supset \blacksquare B \quad (T\text{-export}) \\
\blacksquare A \quad (T\text{-export}) \\
\blacksquare B \\
\vdots \\
(\hat{t}^\blacksquare(\check{k}^\blacksquare M))(\hat{t}^\blacksquare(\check{4}^\blacksquare N))
\end{array}
& \Rightarrow &
\begin{array}{c}
\vdots \\
\blacksquare(\blacksquare A \supset \blacksquare B) \\
\blacksquare A \\
\boxed{\blacksquare} \\
\quad \blacksquare A \supset \blacksquare B \quad (K\text{-import}) \\
\blacksquare A \supset \blacksquare B \quad (T\text{-export}) \\
\blacksquare B \\
\vdots \\
(\hat{t}^\blacksquare(\check{k}^\blacksquare M))N
\end{array}
\end{array}$$

This example shows how the distribution of \hat{t} through a proof term can give rise to more 'global' proof reductions, eliminating detours which are not immediately visible in the natural deduction proof. Since defining proper reduction rules for distribution of \hat{t} over application is technically rather involved, we refer the reader to the discussion of these rules in [Borghuis 1994]. An important conclusion of this discussion is that the set of distribution rules is not as well-behaved as that of annihilations: *Subject Reduction* fails and at best we have *Weak Normalization*.

The point of this section is to show that the 'separation of concerns' that is enforced in Fitch-style systems above K which are extended by rules has advantages for the interpretation of these logics in MPTSs. If all modal steps in the deduction proof are coded by import- and export functions in the proof term, simplifications of the modal structure of the deduction proof can be formalized inside MPTSs in the form of simple subject reduction rules. For systems which are extended by axioms, we can find formulations of some annihilation rules but these are rather awkward compared to those above. Distribution rules cannot be specified for extensions by (T -) axioms.

5. Axioms and rules

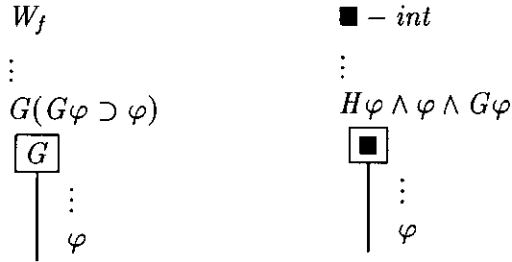
In section 3.3, two ways of extending the Fitch-style deduction system for the basic tense logic K_t were discussed: extension by axioms versus extension by rules. Although it was argued that extension by rules is preferred, there were several tense logical principles for which an extension by axioms was adopted because there didn't seem to be a Fitch-style rule corresponding to them. In this section we take a more systematic look at these cases, to see what we can learn with respect to the general (open) question of expressivity of modal Fitch-style deduction rules.

Throughout this paper, a number of axioms has been discussed for which no corresponding Fitch-style rule was given:

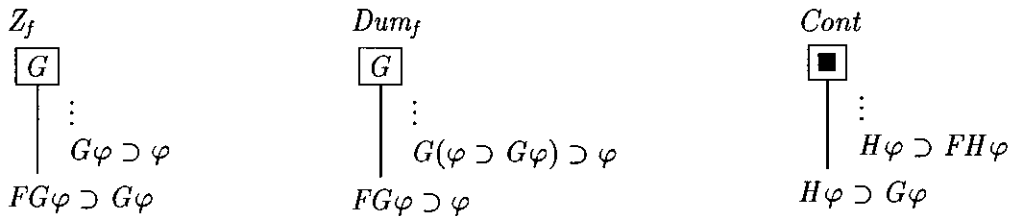
- Z_f $G(G\varphi \supset \varphi) \supset (FG\varphi \supset G\varphi)$ Z_p $H(H\varphi \supset \varphi) \supset (PH\varphi \supset H\varphi)$
- W_f $G(G\varphi \supset \varphi) \supset G\varphi$ W_p $H(H\varphi \supset \varphi) \supset \varphi$
- Dum_f $G(G(\varphi \supset G\varphi) \supset \varphi) \supset (FG\varphi \supset \varphi)$ Dum_p $H(H(\varphi \supset H\varphi) \supset \varphi) \supset (PH\varphi \supset \varphi)$
- $Cont$ $\blacksquare(H\varphi \supset FH\varphi) \supset (H\varphi \supset G\varphi)$
- $\blacksquare-int$ $(H\varphi \wedge \varphi \wedge G\varphi) \supset \blacksquare\varphi$.

However, with a somewhat more insouciant approach to Fitch-style deduction one could come up with rules for these axioms; an import rule for W_f/W_p and $\blacksquare-int$, and an export rule for Z_f/Z_p , Dum_f/Dum_p and $Cont$ (for the axiom pairs, we only show the rule for the f -axiom):

Import rules



Export rules



As the reader can easily check, adding these rules to K_t will make the corresponding axiom a theorem of the resulting system. Vice versa, adding one of the axioms to K_t will make the corresponding rule derivable. The modal rules we had encountered sofar only changed the modality which is the main connective of the formula they import or export. The above rules change modalities *inside* the formulas to which they are applied, and some even change the

propositional form of these formulas. Intuitively such rules are counterproductive; they seem to interfere with the ‘separation of concerns’ they are supposed to promote. This intuition is supported by an observation which has been made both in tense logic and in the meta theory of MPTSs.

In [Prior 1967], Smiley’s ‘proof of consistency’ is mentioned: when a tense logical axiom or rule ‘survives’ the interpretation $G\varphi = H\varphi = F\varphi = P\varphi$, it is deemed consistent. The idea is that an axiom or rule that turns into a tautology when stripped of its tense operators holds in ‘instantaneous time’, where everything happens simultaneously in a single instant. Of the above axioms only W_f and W_p fail Smiley’s test, they turn into $(\varphi \supset \varphi) \supset \varphi$. In the proofs of the meta-theoretical properties of MPTSs ([Borghuis 1994]) this same idea of stripping the modal parts comes up. A vital ingredient of these proofs is a mapping that projects MPTSs back onto their underlying PTS, by erasing everything that is modal in types, terms and contexts.

5.1. DEFINITION. Erasure Mapping

Let $|\cdot|$ be a mapping of MPTS-terms to PTS-terms:

- i. $|\Box^o A| = |A|$, $|\neg\Box^o\neg A| = |A|$ for all $o \in \mathcal{O}$
- ii. $|A_1 A_2| = |A_1| |A_2|$, $|\lambda x : A. b| = \lambda x : |A|. |b|$, $|\Pi x : A. B| = \Pi x : |A|. |B|$
- iii. $|\Gamma, x : A| = |\Gamma|, |x : A|$, $|\mathcal{G} \boxtimes^o \Gamma| = |\mathcal{G}|, |\Gamma|$ for all $o \in \mathcal{O}$
- iv. $|A : B| = |A| : |B|$, $|\varepsilon| = \varepsilon$, $|x| = x$ (for $x \in \text{Var}$), $|s| = s$ (for $s \in \mathcal{S}$)
- v. $|\check{k}^o A| = |A|$, $|\check{b}^o A| = |A|$, $|\check{4}^o A| = |A|$, $|\check{5}^o A| = |A|$,
 $|\hat{k}^o A| = |A|$, $|\hat{d}^o A| = |A|$, $|\hat{t}^o A| = |A|$ for all $o \in \mathcal{O}$.

If we apply this mapping to the standard import and export rules they turn into identities for the underlying PTS, for instance K -import

$$\left| \frac{\mathcal{G} \vdash A : \Box^o B : \text{Prop}}{\mathcal{G} \boxtimes^o \varepsilon \vdash \check{k}^o A : B} \right| = \frac{\Gamma \vdash |A| : |B| : \text{Prop}}{\Gamma \vdash |A| : |B|}$$

(where $\Gamma = |\mathcal{G} \boxtimes^o \varepsilon| = |\mathcal{G}, \varepsilon| = |\mathcal{G}|$). This shows that these modal rules do not in any way *strengthen the underlying non-modal PTS* with respect to the derivation of non-modal statements.

However, all of the deduction rules proposed in this section fail this test, with the exception of the export rules for Z_f/Z_p and Cont . The first example is the type theoretical analogon of W_f -import, which fails the erasure test in the same way that W_f fails Smiley’s test:

$$\left| \frac{\mathcal{G} \vdash M : G(G\varphi \supset \varphi)}{\mathcal{G} \boxtimes^G \varepsilon \vdash \check{w}^G M : \varphi} \right| = \frac{\Gamma \vdash |M| : |\varphi \supset \varphi|}{\Gamma \vdash |M| : |\varphi|}.$$

The stripped rule incorrectly strengthens the PTS by stating that a term $|M|$ of type $|\varphi \supset \varphi|$ on context Γ is also an inhabitant of type $|\varphi|$ on that context. The type theoretical test is stricter than that of Smiley since it demands that the premiss and the consequence of the rule are identical after stripping. This eliminates the rules for the $\text{Dum}_f/\text{Dum}_p$ and \blacksquare -int axioms, which pass Smiley’s test. For instance, the rule for \blacksquare -int:

$$\left| \frac{\mathcal{G} \vdash M : H\varphi \wedge \varphi \wedge G\varphi}{\mathcal{G} \boxtimes^\blacksquare \varepsilon \vdash \check{i}^\blacksquare M : \varphi} \right| = \frac{\Gamma \vdash |M| : |\varphi \wedge \varphi \wedge \varphi|}{\Gamma \vdash |M| : |\varphi|}.$$

Although the type of the premiss statement propositionally implies the type of the conclusion statement, the stripped rule still strengthens the underlying PTS ($\lambda PROP2$). It allows us to immediately use a proof object $|M|$ for $|\varphi \wedge \varphi \wedge \varphi|$ as proof object for $|\varphi|$, whereas the PTS-derivation of a term inhabiting $|\varphi|$ from $|M|$ would require a number of steps changing the form of $|M|$. Hence the erasure mapping leaves us with just the export rules for Z_f/Z_p and $Cont$; the stripped form of all of these rules is the harmless

$$\frac{\Gamma \vdash |M| : |\varphi \supset \varphi|}{\Gamma \vdash |M| : |\varphi \supset \varphi|}.$$

Although these rules seem unproblematic from a type theoretical point of view, they subtly interfere with the separation between modal and propositional steps in Fitch-style proofs. In natural deduction a common ‘bottom up’ strategy for finding a proof for a given formula is to decompose it according to its main connective by assuming that the last step in its proof was the introduction of this connective. This leaves us a deduction problem for a simple formula which can again be decomposed..., etc. In Fitch-style systems for simple modal logics, like K_t , this strategy continues to work: proving a formula of the form $\Box^o \varphi$ can be simplified to proving φ inside a \Box^o -subordinate proof. Rules like those for Z_f/Z_p and $Cont$ undermine such local strategies because they act inside a formula rather than on its main connective. If faced with the task of proving a formula $H\varphi \supset G\varphi$, one would assume that the last step in the proof was the introduction of ‘ \supset ’, rather than reducing the problem to proving $H\varphi \supset FH\varphi$ in a \blacksquare -subordinate proof ($Cont$). In this case a local strategy for finding proofs would suggest a propositional rule where a modal rule is needed. Clearly a further proof-theoretical analysis of modal Fitch-style deduction systems is needed to characterize the tense logical principles of which an extension by rules is possible. However, the above observations suggest that the propositional forms of these principles will be simple.

6. Concluding remarks

This paper shows that the possibilities for temporal reasoning captured by Priorean Tense Logics can be brought into type theory by interpreting these logics in MPTSs. Crucial to this interpretation are the Fitch-style deduction formulations of these logics: the proofs in these modal natural deduction systems correspond directly to term in the MPTSs. Through this close correspondence solutions to problems of temporal reasoning found (in the widespread applications of) tense logics are now available to type theory, which is of particular interest for applications of type theory to knowledge representation.

In section 3.2 a Fitch-style system for the minimal tense logic K_t is given. This system gives a satisfactory deductive account of the fundamental symmetry between the past and future direction of time expressed by the K_t -axioms $\varphi \supset GP\varphi/\varphi \supset HF\varphi$: the effects of this symmetry, both in K_t and its extensions, that were noted in the literature can be reproduced in it. The modal rules of the system suggest why the mirror image property holds for Fitch-style K_t ; for every modal rule for the G -operator there is a ‘mirror image’ rule for the H -operator: K -import and K -export hold for both G and H , and the FB -import rules relating the operators are each others structural mirror image. Hence, given a deduction proof of $\varphi(G, H)$ a proof of $\varphi(H, G)$ can be found by taking the mirror images of all hypotheses in the proof of $\varphi(G, H)$, and matching each application of a modal rule in that proof with an application of the mirror image of that rule. This meta-construction of ‘mirror image’-proofs can be formalized in λK_t (the MPTS corresponding to the deduction system) as a substitution operation on the term representing the proof of $\varphi(G, H)$.

Essentially the Fitch-style system for K_t and λK_t suffice for the type theoretical interpretation of any Priorean Tense Logic, since we can trivially accommodate all further tense logical principles by adding them to the deduction system and the MPTS as (logical) axioms. However, for a number of well-known tense logical principles, such as those expressing transitivity, density, infinity and reflexivity of the flow of time, a more interesting extension of the basic systems is possible. Each of these principles becomes derivable by adding one extra import or export rule to Fitch-style K_t and λK_t (cf. sections 3.3 and 4.3). An advantage of these extensions by rules is that all modal steps in the proof consist in exchanging formulas *between* the main and modal subordinate proofs, whereas all propositional steps take place *inside* the main or modal subordinate proof. This separation of concerns manifests itself in the structure of the MPTS-terms corresponding to these proofs: modal steps are recorded by occurrences of modal functions ($\hat{k}^o, \hat{k}^o, \dots$) in these terms, propositional steps are recorded as λ -abstractions and applications. In proof terms of this kind, certain detours (pointless combinations of modal steps) in natural deduction proofs can be identified as subterms prefixed by specific combinations of modal functions. The removal of these detours, which is a meta-operation on Fitch-style proofs, can be formalized in MPTSs as a set of well-behaved subject reduction rules (see section 4.6).

An interesting family of logics above K_t are the linear tense logics, which are discussed in sections 3.5 and 4.5. Initially, the minimal linear tense logic Lin does not seem amendable to the kind of Fitch-style treatment applied to K_t , because of the form of the characteristic axioms $L_p : P\varphi \supset H(P\varphi \vee \varphi \vee F\varphi)$ and $L_f : F\varphi \supset G(P\varphi \vee \varphi \vee F\varphi)$. However, using the definable universal normal operator ‘ \blacksquare ’ ($\blacksquare\varphi =_{def} H\varphi \wedge \varphi \wedge G\varphi$) these axioms can be rephrased as $L_p : \blacksquare\varphi \supset GH\varphi$ and $L_f : \blacksquare\varphi \supset HG\varphi$ for which perfectly good Fitch-style import rules can be given. All other ways of eliminating the \blacksquare -operator can also be dealt with by modal rules, but its definition cannot be eliminated completely; for some cases of

introduction one half of it is needed, the ‘axiom’ $\blacksquare\text{-int} : H\varphi \wedge \varphi \wedge G\varphi \supset \blacksquare\varphi$. By extending the Fitch-style system for *Lin* and the corresponding MPTS λLin with further axioms and rules, tense logics describing familiar linear conceptions of time can be captured type theoretically. From a deductive point of view the systems for ‘rational time’ $((Q, <))$ and $((Q, \leq))$ are of particular interest, since they can be obtained from *Lin* and λLin by extending only with rules. The discussion in section 5 shows that in general the prospect of capturing a tense logic deductively by extending the Fitch-style system for K_t with only rules, is limited to tense logics which have syntactically simple axioms.

Acknowledgements

The author would like to thank Rob Nederpelt for valuable comments on draft versions of this paper, and Anita Klooster for typesetting the manuscript in \LaTeX . The natural deduction proofs in this paper were produced using the ‘Macro’s voor het zetten van Fitch bewijzen’ of R. Sommerhalder.

References

- [Ahn et al. 1994] Ahn, R.M.C., Beun, R.J., Borghuis, T., Bunt, H.C. and van Overveld, C.W.A.M., The DenK-architecture: a fundamental approach to user-interfaces. In *Artificial Intelligence Review* 8, 1994-5, Kluwer Academic Publishers, The Netherlands (pp. 431-455).
- [Barendregt 1992] Barendregt, Henk, Lambda calculi with types. In *Handbook of logic in computer science*, Abramsky, Gabbay and Maibaum (eds.), Oxford University Press, Oxford.
- [van Benthem 1983] van Benthem, J.F.A.K., *The logic of time*, Reidel, Dordrecht.
- [Borghuis 1994] Borghuis, Tijn, *Coming to Terms with Modal Logic*. Ph.D. thesis, Eindhoven University of Technology.
- [Burgess 1984] Burgess, John, P., Basic tense logic. In *Handbook of philosophical logic* Vol. II, Gabbay and Guenther (eds.), Reidel, Dordrecht (pp. 89-133).
- [Chellas 1980] Chellas, Brian F., *Modal logic: an introduction*. Cambridge University Press, Cambridge.
- [Fitch 1952] Fitch, Frederic Brenton, *Symbolic logic, an introduction*. The Ronald Press Company, New York.
- [Gabbay et al. 1994] Gabbay, Dov, Hodkinson, Ian, Reynolds, Mark, *Temporal Logic, Mathematical Foundations and Computational Aspects*, Volume 1. Clarendon Press, Oxford.
- [Geuvers 1993] Geuvers, Herman, *Logics and type systems*. Ph.D. thesis, University of Nijmegen.
- [Gamut 1982] Gamut, L.T.F., *Logica, taal en betekenis 2*. Spectrum, Utrecht.
- [Goldblatt 1992] Goldblatt, Robert, *Logics of time and computation*. CSLI, Stanford.
- [van der Hoek 1992] van der Hoek, Wiebe, *Modalities for reasoning about knowledge and quantities*, Ph.D. thesis, Free University of Amsterdam, Amsterdam.
- [Hughes and Cresswell 1972] Hughes, G.E. and Cresswell, M.J., *An introduction to modal logic*. University Paperback, London.
- [Koymans 1989] Koymans, Ron, *Specifying Message Passing and Time-Critical Systems with Temporal Logic*, Ph.D. thesis, Eindhoven University of Technology, Eindhoven.

- [Prior 1957] Prior, A.N., *Time and Modality*, Clarendon Press, Oxford.
- [Prior 1967] Prior, A.N., *Past, present and future*, Clarendon Press, Oxford.
- [van Westrhenen et al. 1993] van Westrhenen, S.C., Sommerhalder, R., Tonino, J.F.M., *LOGICA een inleiding met toepassingen in de informatica*, Academic Service , Schoonhoven.

Appendix: Interderivability of $\Diamond A$ and $\neg\Box\neg A$

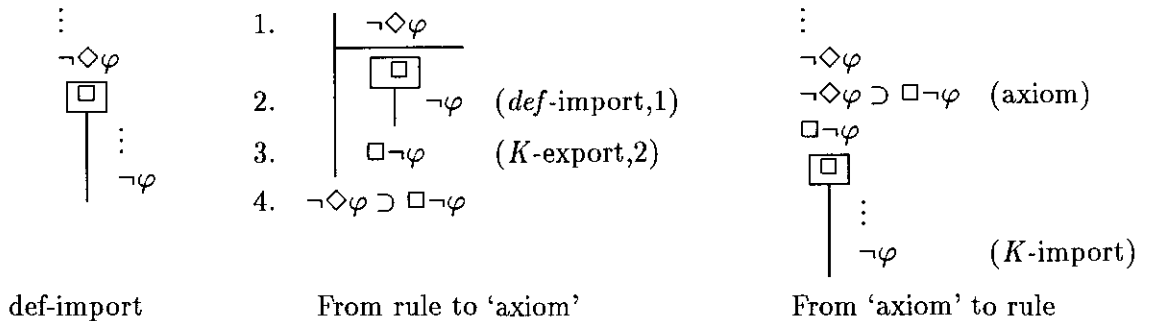
Throughout this paper, all natural deduction rules and type derivation rules were stated using only the ‘universal’ (or ‘ \Box ’-) modalities ‘ G ’ and ‘ H ’. The ‘existential’ (or ‘ \Diamond ’-) operators ‘ F ’ and ‘ P ’ were treated as definitional abbreviations of ‘ $\neg G\neg$ ’ and ‘ $\neg H\neg$ ’ respectively. This is sufficient for the purpose of this paper, but one may prefer a deduction or derivation system in which ‘ F ’ and ‘ P ’ are first-class citizens, bringing the proofs closer to the standard presentation of the axiomatics and model theory of tense logics in the literature. However, in a system that has both universal and existential operators the equivalences $F\varphi \leftrightarrow \neg G\neg\varphi$ and $P\varphi \leftrightarrow \neg H\neg\varphi$ (in general $\Diamond\varphi \leftrightarrow \neg\Box\neg\varphi$) should be *derivable* instead of dependent on a definition.

The easiest way to bring the definitions into the deduction proofs would be to introduce rules which allow the replacement of an occurrence of the existential operator ‘ \Diamond ’ with that of an occurrence of ‘ $\neg\Box\neg$ ’ (and vice versa) in a single step:

$$\begin{array}{c} \vdots \\ \Diamond\varphi \\ \neg\Box\neg\varphi \end{array} \qquad \begin{array}{c} \vdots \\ \neg\Box\neg\varphi \\ \Diamond\varphi \end{array}$$

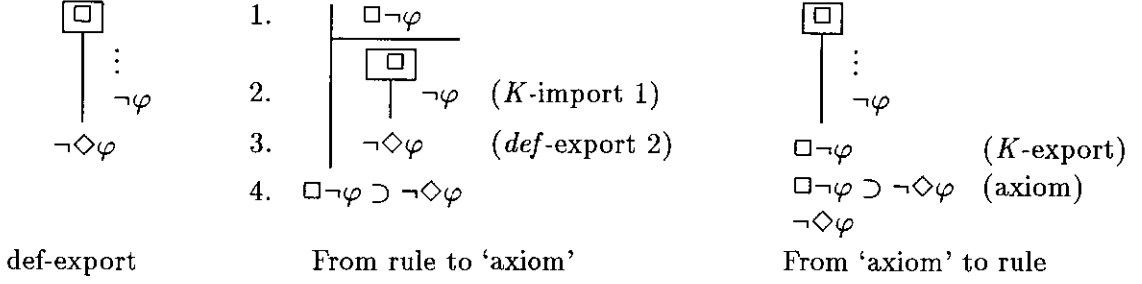
Although completely straightforward, this solution is not in the spirit of the ‘separation of concerns’ advocated earlier. According to this idea propositional reasoning steps in a natural deduction proof are carried out inside hypothesis-intervals of the same modal depth, whereas modal reasoning steps correspond to the transfer of formulas between hypothesis-intervals of different modal depth. The ‘replacement rules’ given above do not respect this distinction: they code a modal equivalence by means of rules that are carried out inside a single hypothesis-interval.

If we state the relation between the universal and the existential operator as $\neg\Diamond\varphi \leftrightarrow \Box\neg\varphi$, a solution can be found that is more in line with what we have done sofar. Given the K -deduction rules for ‘ \Box ’, this equivalence becomes derivable by adding an extra import rule (def-import) for the direction $\neg\Diamond\varphi \supset \Box\neg\varphi$ and an extra export-rule (def-export) for the direction $\Box\neg\varphi \supset \neg\Diamond\varphi$.

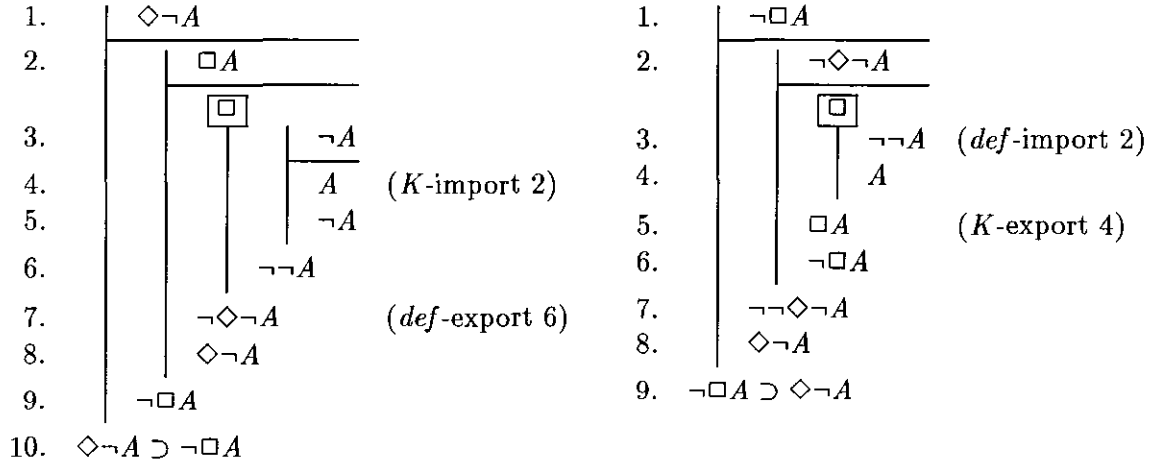


The intuition behind the rule is that if $\neg\Diamond\varphi$ holds there is no accessible world (future/past time point) when φ holds, hence $\neg\varphi$ has to hold in every accessible world (future/past time point), i.e. $\neg\varphi$ is true in the arbitrary accessible world represented by the strict subordinate

proof. The export rule formalizes the converse intuition: if $\neg\varphi$ holds in an arbitrary world, φ will not be true in any accessible world and hence $\neg\Diamond\varphi$ holds:



To show that these two rules are sufficient, we give the deduction proof for the remaining cases covered by the definition of ‘ \Diamond ’ by means of ‘ \square ’:

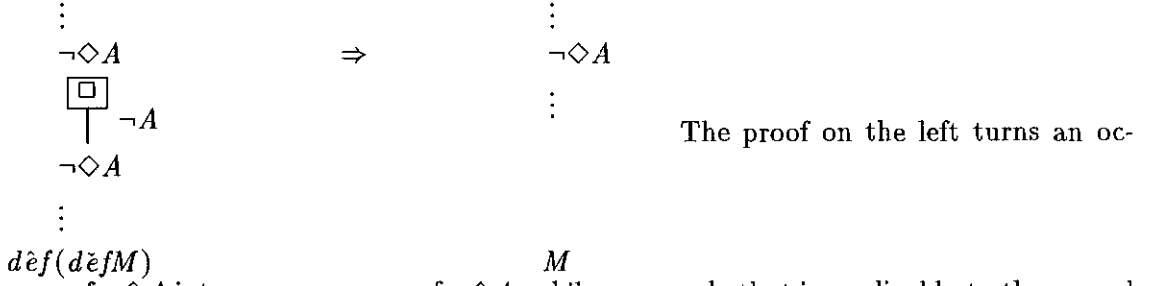


This strategy for bringing the definition into the natural deduction system can also be applied to MPTSs for tense logics. All that is required for a type theoretical translation of *def-import* and *-export* is a slight extension of the syntax: we have to allow ‘existential modal types’ ($\Diamond\mathcal{T}$), and new functions ($d\check{e}f, d\hat{e}f$) that will record the use of the ‘definition-rules’ in the proof object.

$$\text{def-import} \quad \frac{\mathcal{G} \vdash A : \neg\Diamond B : Prop}{\mathcal{G} \boxplus \varepsilon \vdash d\check{e}fA : \neg B} \qquad \text{def-export} \quad \frac{\mathcal{G} \boxplus \varepsilon \vdash A : \neg B : Prop}{\mathcal{G} \vdash d\hat{e}fA : \neg\Diamond B} .$$

Clearly, these rules enable us to transform any ‘ \square -type’ statement into its ‘ \Diamond -type’ counterpart (and vice versa), by means of derivations analogous to the natural deduction proofs above.

A disadvantage of incorporating the relation between ‘ \square ’ and ‘ \Diamond ’ by means of deduction (or derivation) rules is that proofs (derivations) can turn out to contain pointless combinations of steps involving the definition rules. The simplest example of such a detour is the subsequent application of *def-import* and *def-export* to a formula of the form $\neg\Diamond A$:

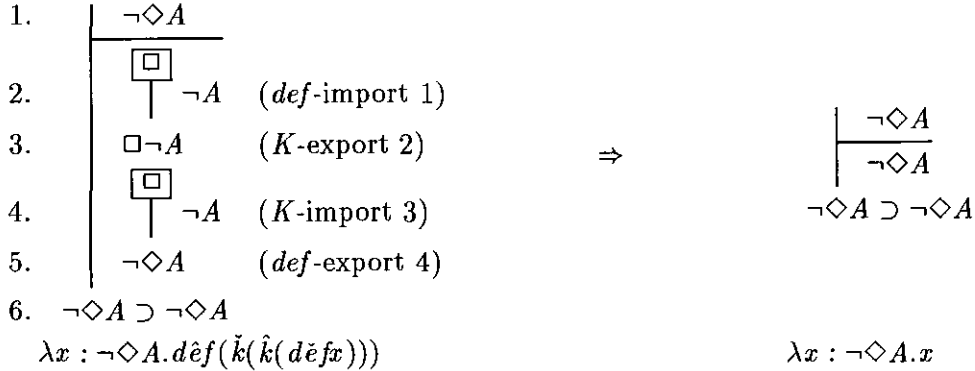


The proof on the left turns an oc-

currence of $\neg\Diamond A$ into an occurrence of $\neg\Diamond A$, while every rule that is applicable to the second occurrence was already applicable to the first occurrence. Obviously, this proof could be simplified to the one on the right by omitting the detour consisting of *def*-import and *def*-export. In natural deduction such simplifications are meta-operations on proofs but, as pointed out earlier, type theoretically they can sometimes be expressed as subject reduction rules. The signature of the detour in the above proof is a subterm of the form $d\hat{e}f(d\check{e}fM)$, where M represents the proof upto the first occurrence of $\neg\Diamond A$. Since the signature of the simplified proof is M , the situation seems to be analogous to that for *K*-import and *K*-export discussed earlier. Hence we stipulate annihilation for the definition rules:

$$\begin{aligned}
d\hat{e}f(d\check{e}fM) &\Rightarrow M \\
d\check{e}f(d\hat{e}fM) &\Rightarrow M .
\end{aligned}$$

In tandem with the annihilations for *K*-import and -export, these rules are able to eliminate more interesting detours than the one in the example above:



The proof on the left derives $\neg\Diamond A$ from $\neg\Diamond A$ via $\square\neg A$ instead of establishing this identity directly as in the proof on the right. In the corresponding proof term this detour is recorded as a sequence of *K*- and *def*-functions which can be reduced in the following way:

$$\begin{aligned}
\lambda x : \neg\Diamond A. d\hat{e}f(\check{k}(\hat{k}(d\check{e}fx))) &\rightarrow_{\check{k}\hat{k}} \\
\lambda x : \neg\Diamond A. d\hat{e}f(d\check{e}fx) &\rightarrow_{d\check{e}fd\check{e}f} \\
\lambda x : \neg\Diamond A. x &.
\end{aligned}$$

The resulting proof term is the identity function for $x : \neg\Diamond A$, which corresponds to the proof on the right; the combined reduction rules have eliminated the detour. In the same way a proof of $\square\neg A$ from $\square\neg A$ via $\neg\Diamond A$ can be reduced to an identity proof. However, these examples should not lead us to believe that any ‘unnecessary’ application of the definition rules in a proof can be eliminated using *K*-annihilation, *def*-annihilation and β -reduction:

the proofs of $\diamond\neg A$ from $\diamond\neg A$ through $\neg\Box A$ and $\neg\Box A$ from $\neg\Box A$ via $\diamond\neg A$ do not reduce to identity. This is caused by the application of $\neg\neg$ -elimination in these proofs (cf. the deduction proof of $\neg\Box A \supset \diamond\neg A$ given earlier). Since MPTSs are by heritage intuitionistic, double negation-elimination requires invoking the logical axiom $c : \forall\alpha : Prop(\neg\neg\alpha \supset \alpha)$. This leaves a ‘scar’ in the proof object (consisting of the constant c applied to some proposition type) which blocks the reduction path leading to the identity function.

The discussion of import- and export-rules as a means for expressing the definition of ‘ \diamond ’ as ‘ $\neg\Box\neg$ ’ was started in an attempt to answer the practical question how a deduction or type system could be obtained in which the operators ‘ F ’ and ‘ P ’ are first-class citizens. The practical conclusions of the discussion are that for the natural deduction system we only have to add the *def*-import and *def*-export rules relating G to F and H to P , whereas the MPTS ($\lambda\Box PROP2$) needs to be extended with a couple of things.

- ‘Boxing rules’ for F and P , stating that every proposition type may be prefixed with the existential modal operators:

$$\text{Boxing}_F \frac{\mathcal{G} \vdash A : Prop}{\mathcal{G} \vdash FA : Prop} \quad \text{Boxing}_P \frac{\mathcal{G} \vdash A : Prop}{\mathcal{G} \vdash PA : Prop} .$$

- *def*-import and -export rules relating F to G and P to H

$$\begin{array}{ll} \text{def-import}_G \frac{\mathcal{G} \vdash A : \neg FB : Prop}{\mathcal{G} \boxtimes_G \varepsilon \vdash d\check{e}f_G A : \neg B} & \text{def-export}_G \frac{\mathcal{G} \boxtimes_G \varepsilon \vdash A : \neg B : Prop}{\mathcal{G} \vdash d\hat{e}f_G A : \neg FB} \\ \text{def-import}_H \frac{\mathcal{G} \vdash A : \neg PB : Prop}{\mathcal{G} \boxtimes_H \varepsilon \vdash d\check{e}f_H A : \neg FB} & \text{def-export}_H \frac{\mathcal{G} \boxtimes_H \varepsilon \vdash A : \neg B : Prop}{\mathcal{G} \vdash d\hat{e}f_H A : \neg PB} . \end{array}$$

- Annihilation rules for *def*-import/export combinations (optional)

$$\begin{array}{ll} d\check{e}f_G(d\hat{e}f_G M) \Rightarrow M & d\hat{e}f_G(d\check{e}f_G M) \Rightarrow M \\ d\check{e}f_H(d\hat{e}f_H M) \Rightarrow M & d\hat{e}f_H(d\check{e}f_H M) \Rightarrow M . \end{array}$$

The additional Boxing- and import/export rules have no influence on the meta-theoretical properties of the MPTS, the proofs of these properties in [Borghuis 1994] can be extended to deal with these rules in a straightforward way. Similarly, the well-behavedness of the *dēf* & *dēf* annihilation (by itself and in combination with the K -annihilation) can be shown by adapting the proofs for Subject Reduction, Strong Normalization and Church Rosser given those for the \check{k} & \hat{k} annihilation.

In this series appeared:

| | | |
|-------|--------------------------------------------------|---------------------------------------------------------------------------------------------------------|
| 93/01 | R. van Geldrop | Deriving the Aho-Corasick algorithms: a case study into the synergy of programming methods, p. 36. |
| 93/02 | T. Verhoeff | A continuous version of the Prisoner's Dilemma, p. 17 |
| 93/03 | T. Verhoeff | Quicksort for linked lists, p. 8. |
| 93/04 | E.H.L. Aarts J.H.M. Korst P.J. Zwietering | Deterministic and randomized local search, p. 78. |
| 93/05 | J.C.M. Baeten C. Verhoef | A congruence theorem for structured operational semantics with predicates, p. 18. |
| 93/06 | J.P. Veltkamp | On the unavoidability of metastable behaviour, p. 29 |
| 93/07 | P.D. Moerland | Exercises in Multiprogramming, p. 97 |
| 93/08 | J. Verhoosel | A Formal Deterministic Scheduling Model for Hard Real-Time Executions in DEDOS, p. 32. |
| 93/09 | K.M. van Hee | Systems Engineering: a Formal Approach Part I: System Concepts, p. 72. |
| 93/10 | K.M. van Hee | Systems Engineering: a Formal Approach Part II: Frameworks, p. 44. |
| 93/11 | K.M. van Hee | Systems Engineering: a Formal Approach Part III: Modeling Methods, p. 101. |
| 93/12 | K.M. van Hee | Systems Engineering: a Formal Approach Part IV: Analysis Methods, p. 63. |
| 93/13 | K.M. van Hee | Systems Engineering: a Formal Approach Part V: Specification Language, p. 89. |
| 93/14 | J.C.M. Baeten J.A. Bergstra | On Sequential Composition, Action Prefixes and Process Prefix, p. 21. |
| 93/15 | J.C.M. Baeten J.A. Bergstra R.N. Bol | A Real-Time Process Logic, p. 31. |
| 93/16 | H. Schepers J. Hooman | A Trace-Based Compositional Proof Theory for Fault Tolerant Distributed Systems, p. 27 |
| 93/17 | D. Alstein P. van der Stok | Hard Real-Time Reliable Multicast in the DEDOS system, p. 19. |
| 93/18 | C. Verhoef | A congruence theorem for structured operational semantics with predicates and negative premises, p. 22. |
| 93/19 | G-J. Houben | The Design of an Online Help Facility for ExSpect, p.21. |
| 93/20 | F.S. de Boer | A Process Algebra of Concurrent Constraint Programming, p. 15. |
| 93/21 | M. Codish D. Dams G. Filé M. Bruynooghe | Freeness Analysis for Logic Programs - And Correctness, p. 24 |
| 93/22 | E. Poll | A Typechecker for Bijective Pure Type Systems, p. 28. |
| 93/23 | E. de Kogel | Relational Algebra and Equational Proofs, p. 23. |
| 93/24 | E. Poll and Paula Severi | Pure Type Systems with Definitions, p. 38. |
| 93/25 | H. Schepers and R. Gerth | A Compositional Proof Theory for Fault Tolerant Real-Time Distributed Systems, p. 31. |
| 93/26 | W.M.P. van der Aalst | Multi-dimensional Petri nets, p. 25. |
| 93/27 | T. Kloks and D. Kratsch | Finding all minimal separators of a graph, p. 11. |
| 93/28 | F. Kamareddine and R. Nederpelt | A Semantics for a fine λ -calculus with de Bruijn indices, p. 49. |
| 93/29 | R. Post and P. De Bra | GOLD, a Graph Oriented Language for Databases, p. 42. |
| 93/30 | J. Deogun T. Kloks D. Kratsch H. Müller | On Vertex Ranking for Permutation and Other Graphs, p. 11. |

| | | |
|-------|--------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| 93/31 | W. Körver | Derivation of delay insensitive and speed independent CMOS circuits, using directed commands and production rule sets, p. 40. |
| 93/32 | H. ten Eikelder and H. van Geldrop | On the Correctness of some Algorithms to generate Finite Automata for Regular Expressions, p. 17. |
| 93/33 | L. Loyens and J. Moonen | ILIAS, a sequential language for parallel matrix computations, p. 20. |
| 93/34 | J.C.M. Baeten and J.A. Bergstra | Real Time Process Algebra with Infinitesimals, p.39. |
| 93/35 | W. Ferrer and P. Severi | Abstract Reduction and Topology, p. 28. |
| 93/36 | J.C.M. Baeten and J.A. Bergstra | Non Interleaving Process Algebra, p. 17. |
| 93/37 | J. Brunekreef J-P. Katoen R. Koymans S. Mauw | Design and Analysis of Dynamic Leader Election Protocols in Broadcast Networks, p. 73. |
| 93/38 | C. Verhoef | A general conservative extension theorem in process algebra, p. 17. |
| 93/39 | W.P.M. Nuijten E.H.L. Aarts D.A.A. van Erp Taalman Kip K.M. van Hee | Job Shop Scheduling by Constraint Satisfaction, p. 22. |
| 93/40 | P.D.V. van der Stok M.M.M.P.J. Claessen D. Alstein | A Hierarchical Membership Protocol for Synchronous Distributed Systems, p. 43. |
| 93/41 | A. Bijlsma | Temporal operators viewed as predicate transformers, p. 11. |
| 93/42 | P.M.P. Rambags | Automatic Verification of Regular Protocols in P/T Nets, p. 23. |
| 93/43 | B.W. Watson | A taxonomy of finite automata construction algorithms, p. 87. |
| 93/44 | B.W. Watson | A taxonomy of finite automata minimization algorithms, p. 23. |
| 93/45 | E.J. Luit J.M.M. Martin | A precise clock synchronization protocol,p. |
| 93/46 | T. Kloks D. Kratsch J. Spinrad | Treewidth and Patwidth of Cocomparability graphs of Bounded Dimension, p. 14. |
| 93/47 | W. v.d. Aalst P. De Bra G.J. Houben Y. Komatzky | Browsing Semantics in the "Tower" Model, p. 19. |
| 93/48 | R. Gerth | Verifying Sequentially Consistent Memory using Interface Refinement, p. 20. |
| 94/01 | P. America M. van der Kammen R.P. Nederpelt O.S. van Roosmalen H.C.M. de Swart | The object-oriented paradigm, p. 28. |
| 94/02 | F. Kamareddine R.P. Nederpelt | Canonical typing and Π -conversion, p. 51. |
| 94/03 | L.B. Hartman K.M. van Hee | Application of Marcov Decision Prozesse to Search Problems, p. 21. |
| 94/04 | J.C.M. Baeten J.A. Bergstra | Graph Isomorphism Models for Non Interleaving Process Algebra, p. 18. |
| 94/05 | P. Zhou J. Hooman | Formal Specification and Compositional Verification of an Atomic Broadcast Protocol, p. 22. |
| 94/06 | T. Basten T. Kunz J. Black M. Coffin D. Taylor | Time and the Order of Abstract Events in Distributed Computations, p. 29. |
| 94/07 | K.R. Apt R. Bol | Logic Programming and Negation: A Survey, p. 62. |
| 94/08 | O.S. van Roosmalen | A Hierarchical Diagrammatic Representation of Class Structure, p. 22. |
| 94/09 | J.C.M. Baeten J.A. Bergstra | Process Algebra with Partial Choice, p. 16. |

| | | |
|-------|--------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|
| 94/10 | T. verhoeff | The testing Paradigm Applied to Network Structure. p. 31. |
| 94/11 | J. Peleska C. Huizing C. Petersohn | A Comparison of Ward & Mellor's Transformation Schema with State- & Activitycharts, p. 30. |
| 94/12 | T. Kloks D. Kratsch H. Müller | Dominoes, p. 14. |
| 94/13 | R. Seljée | A New Method for Integrity Constraint checking in Deductive Databases, p. 34. |
| 94/14 | W. Peremans | Ups and Downs of Type Theory, p. 9. |
| 94/15 | R.J.M. Vaessens E.H.L. Aarts J.K. Lenstra | Job Shop Scheduling by Local Search, p. 21. |
| 94/16 | R.C. Backhouse H. Doornbos | Mathematical Induction Made Computational, p. 36. |
| 94/17 | S. Mauw M.A. Reniers | An Algebraic Semantics of Basic Message Sequence Charts, p. 9. |
| 94/18 | F. Kamareddine R. Nederpelt | Refining Reduction in the Lambda Calculus, p. 15. |
| 94/19 | B.W. Watson | The performance of single-keyword and multiple-keyword pattern matching algorithms, p. 46. |
| 94/20 | R. Bloo F. Kamareddine R. Nederpelt | Beyond β -Reduction in Church's $\lambda \rightarrow$, p. 22. |
| 94/21 | B.W. Watson | An introduction to the Fire engine: A C++ toolkit for Finite automata and Regular Expressions. |
| 94/22 | B.W. Watson | The design and implementation of the FIRE engine: A C++ toolkit for Finite automata and regular Expressions. |
| 94/23 | S. Mauw and M.A. Reniers | An algebraic semantics of Message Sequence Charts, p. 43. |
| 94/24 | D. Dams O. Grumberg R. Gerth | Abstract Interpretation of Reactive Systems: Abstractions Preserving \forall CTL*, \exists CTL* and CTL*, p. 28. |
| 94/25 | T. Kloks | $K_{1,3}$ -free and W_4 -free graphs, p. 10. |
| 94/26 | R.R. Hoogerwoord | On the foundations of functional programming: a programmer's point of view, p. 54. |
| 94/27 | S. Mauw and H. Mulder | Regularity of BPA-Systems is Decidable, p. 14. |
| 94/28 | C.W.A.M. van Overveld M. Verhoeven | Stars or Stripes: a comparative study of finite and transfinite techniques for surface modelling, p. 20. |
| 94/29 | J. Hooman | Correctness of Real Time Systems by Construction, p. 22. |
| 94/30 | J.C.M. Baeten J.A. Bergstra Gh. Ştefanescu | Process Algebra with Feedback, p. 22. |
| 94/31 | B.W. Watson R.E. Watson | A Boyer-Moore type algorithm for regular expression pattern matching, p. 22. |
| 94/32 | J.J. Vereijken | Fischer's Protocol in Timed Process Algebra, p. 38. |
| 94/33 | T. Laan | A formalization of the Ramified Type Theory, p.40. |
| 94/34 | R. Bloo F. Kamareddine R. Nederpelt | The Barendregt Cube with Definitions and Generalised Reduction, p. 37. |
| 94/35 | J.C.M. Baeten S. Mauw | Delayed choice: an operator for joining Message Sequence Charts, p. 15. |
| 94/36 | F. Kamareddine R. Nederpelt | Canonical typing and Π -conversion in the Barendregt Cube, p. 19. |
| 94/37 | T. Basten R. Bol M. Voorhoeve | Simulating and Analyzing Railway Interlockings in ExSpect, p. 30. |
| 94/38 | A. Bijlsma C.S. Scholten | Point-free substitution, p. 10. |

| | | | |
|-------|--------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------|--------------------------------------------------|
| 94/39 | A. Blokhuis T. Kloks | On the equivalence covering number of splitgraphs, p. 4. | |
| 94/40 | D. Alstein | Distributed Consensus and Hard Real-Time Systems, p. 34. | |
| 94/41 | T. Kloks D. Kratsch | Computing a perfect edge without vertex elimination ordering of a chordal bipartite graph, p. 6. | |
| 94/42 | J. Engelfriet J.J. Vereijken | Concatenation of Graphs, p. 7. | |
| 94/43 | R.C. Backhouse M. Bijsterveld | Category Theory as Coherently Constructive Lattice Theory: An Illustration, p. 35. | |
| 94/44 | E. Brinksmma R. Gerth W. Janssen S. Katz M. Poel C. Rump | J. Davies S. Graf B. Jonsson G. Lowe A. Pnueli J. Zwiers | Verifying Sequentially Consistent Memory, p. 160 |
| 94/45 | G.J. Houben | Tutorial voor de ExSpec-bibliotheek voor "Administratieve Logistiek", p. 43. | |
| 94/46 | R. Bloo F. Kamareddine R. Nederpelt | The λ -cube with classes of terms modulo conversion, p. 16. | |
| 94/47 | R. Bloo F. Kamareddine R. Nederpelt | On Π -conversion in Type Theory, p. 12. | |
| 94/48 | Mathematics of Program Construction Group | Fixed-Point Calculus, p. 11. | |
| 94/49 | J.C.M. Baeten J.A. Bergstra | Process Algebra with Propositional Signals, p. 25. | |
| 94/50 | H. Geuvers | A short and flexible proof of Strong Normalization for the Calculus of Constructions, p. 27. | |
| 94/51 | T. Kloks D. Kratsch H. Müller | Listing simplicial vertices and recognizing diamond-free graphs, p. 4. | |
| 94/52 | W. Penczek R. Kuiper | Traces and Logic, p. 81 | |
| 94/53 | R. Gerth R. Kuiper D. Peled W. Penczek | A Partial Order Approach to Branching Time Logic Model Checking, p. 20. | |
| 95/01 | J.J. Lukkien | The Construction of a small CommunicationLibrary, p.16. | |
| 95/02 | M. Bezem R. Bol J.F. Grooten | Formalizing Process Algebraic Verifications in the Calculus of Constructions, p.49. | |
| 95/03 | J.C.M. Baeten C. Verhoef | Concrete process algebra, p. 134. | |
| 95/04 | J. Hidders | An Isotopic Invariant for Planar Drawings of Connected Planar Graphs, p. 9. | |
| 95/05 | P. Severi | A Type Inference Algorithm for Pure Type Systems, p.20. | |
| 95/06 | T.W.M. Vossen M.G.A. Verhoeven H.M.M. ten Eikelder E.H.L. Aarts | A Quantitative Analysis of Iterated Local Search, p.23. | |
| 95/07 | G.A.M. de Bruyn O.S. van Roosmalen | Drawing Execution Graphs by Parsing, p. 10. | |
| 95/08 | R. Bloo | Preservation of Strong Normalisation for Explicit Substitution, p. 12. | |
| 95/09 | J.C.M. Baeten J.A. Bergstra | Discrete Time Process Algebra, p. 20 | |
| 95/10 | R.C. Backhouse R. Verhoeven O. Weber | MathPad: A System for On-Line Preparation of Mathematical Documents, p. 15 | |

| | | |
|-------|-----------------------------------------------|----------------------------------------------------------------------------------------------------------|
| 95/11 | R. Seljée | Deductive Database Systems and integrity constraint checking, p. 36. |
| 95/12 | S. Mauw and M. Reniers | Empty Interworkings and Refinement Semantics of Interworkings Revised, p. 19. |
| 95/13 | B.W. Watson and G. Zwaan | A taxonomy of sublinear multiple keyword pattern matching algorithms, p. 26. |
| 95/14 | A. Ponse, C. Verhoef, S.F.M. Vlijmen (eds.) | De proceedings: ACP'95, p. |
| 95/15 | P. Niebert and W. Penczek | On the Connection of Partial Order Logics and Partial Order Reduction Methods, p. 12. |
| 95/16 | D. Dams, O. Grumberg, R. Gerth | Abstract Interpretation of Reactive Systems: Preservation of CTL*, p. 27. |
| 95/17 | S. Mauw and E.A. van der Meulen | Specification of tools for Message Sequence Charts, p. 36. |
| 95/18 | F. Kamareddine and T. Laan | A Reflection on Russell's Ramified Types and Kripke's Hierarchy of Truths, p. 14. |
| 95/19 | J.C.M. Baeten and J.A. Bergstra | Discrete Time Process Algebra with Abstraction, p. 15. |
| 95/20 | F. van Raamsdonk and P. Severi | On Normalisation, p. 33. |
| 95/21 | A. van Deursen | Axiomatizing Early and Late Input by Variable Elimination, p. 44. |
| 95/22 | B. Arnold, A. v. Deursen, M. Res | An Algebraic Specification of a Language for Describing Financial Products, p. 11. |
| 95/23 | W.M.P. van der Aalst | Petri net based scheduling, p. 20. |
| 95/24 | F.P.M. Dignum, W.P.M. Nuijten, L.M.A. Janssen | Solving a Time Tabling Problem by Constraint Satisfaction, p. 14. |
| 95/25 | L. Feijs | Synchronous Sequence Charts In Action, p. 36. |
| 95/26 | W.M.P. van der Aalst | A Class of Petri nets for modeling and analyzing business processes, p. 24. |
| 95/27 | P.D.V. van der Stok, J. van der Wal | Proceedings of the Real-Time Database Workshop, p. 106. |
| 95/28 | W. Fokkink, C. Verhoef | A Conservative Look at term Deduction Systems with Variable Binding, p. 29. |
| 95/29 | H. Jurjus | On Nesting of a Nonmonotonic Conditional, p. 14 |
| 95/30 | J. Hidders, C. Hoskens, J. Paredaens | The Formal Model of a Pattern Browsing Technique, p.24. |
| 95/31 | P. Kelb, D. Dams and R. Gerth | Practical Symbolic Model Checking of the full μ -calculus using Compositional Abstractions, p. 17. |
| 95/32 | W.M.P. van der Aalst | Handboek simulatie, p. 51. |
| 95/33 | J. Engelfriet and JJ. Vereijken | Context-Free Graph Grammars and Concatenation of Graphs, p. 35. |
| 95/34 | J. Zwanenburg | Record concatenation with intersection types, p. 46. |
| 95/35 | T. Basten and M. Voorhoeve | An algebraic semantics for hierarchical P/T Nets, p. 32. |
| 96/01 | M. Voorhoeve and T. Basten | Process Algebra with Autonomous Actions, p. 12. |
| 96/02 | P. de Bra and A. Aerts | Multi-User Publishing in the Web: DreSS, A Document Repository Service Station, p. 12 |
| 96/03 | W.M.P. van der Aalst | Parallel Computation of Reachable Dead States in a Free-choice Petri Net, p. 26. |
| 96/04 | S. Mauw | Example specifications in phi-SDL. |
| 96/05 | T. Basten and W.M.P. v.d. Aalst | A Process-Algebraic Approach to Life-Cycle Inheritance Inheritance = Encapsulation + Abstraction, p. 15. |
| 96/06 | W.M.P. van der Aalst and T. Basten | Life-Cycle Inheritance A Petri-Net-Based Approach, p. 18. |
| 96/07 | M. Voorhoeve | Structural Petri Net Equivalence, p. 16. |
| 96/08 | A.T.M. Aerts, P.M.E. De Bra, J.T. de Munk | OODB Support for WWW Applications: Disclosing the internal structure of Hyperdocuments, p. 14. |
| 96/09 | F. Dignum, H. Weigand, E. Verharen | A Formal Specification of Deadlines using Dynamic Deontic Logic, p. 18. |
| 96/10 | R. Bloo, H. Geuvers | Explicit Substitution: on the Edge of Strong Normalisation, p. 13. |
| 96/11 | T. Laan | AUTOMATH and Pure Type Systems, p. 30. |
| 96/12 | F. Kamareddine and T. Laan | A Correspondence between Nuprl and the Ramified Theory of Types, p. 12. |