Coming to terms with modal logic : on the interpretation of modalities in typed lambda-calculus

DOI:
10.6100/IR427575

Document status and date:
Published: 01/01/1994

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

Coming to Terms with Modal Logic: On the interpretation of modalities in typed λ-calculus

Tijn Borghuis

Coming to Terms with Modal Logic:

On the Interpretation of Modalities in Typed λ-Calculus

Proefschrift

ter verkrijging van de graad van doctor aan de Technische Universiteit Eindhoven, op gezag van de Rector Magnificus, prof.dr. J.H. van Lint, voor een commissie aangewezen door het College van Dekanen in het openbaar te verdedigen op vrijdag 9 december 1994 om 16.00 uur

door

Valentijn Anton Johan Borghuis

geboren te Oldenzaal

Dit proefschrift is goedgekeurd door de promotoren

prof.dr. J. C. M. Baeten
prof.dr. H. C. Bunt
en de copromotor
dr. R. P. Nederpelt

Contents

Preface

In one sentence the aim of this thesis is to incorporate intensional reasoning in type theory by means of a 'propositions-as-types' interpretation of modal logic.

To explain this goal properly, we first introduce the reader briefly to the traditional setting of this monograph: the embedding of logics in type theory. These embeddings have been studied extensively, motivated by applications in proof theory, mathematics and computer science. After a brief survey of this research, we focus on the recent use of type theory in 'knowledge representation' which motivated our study of the propositions-as-types interpretation of modal logic. Then the objectives of this thesis are stated, followed by an explanation of the choice of formal frameworks. We conclude by giving an overview of the contents of the thesis.

The propositions-as-types interpretation

The language of typed lambda calculus consists of 'statements' which are of the general form $A : B$, expressing that the 'term' A is of 'type' B (also pronounced as 'object A has type B', or 'A is an inhabitant of B'). Whether such a statement on the relation between expressions A and B is correct, has to be decided in relation to the 'context'. A context is an ordered sequence of statements, which contains all relevant information about the objects and types that are considered primitive. To show that a term (A) of a certain type (B) exists, one has to demonstrate that a statement $(A : B)$ to this effect is derivable on a given context. Generally speaking, this involves showing that the statement is either already present in the context, or can be obtained from the statements in the context by means of the 'derivation rules'. These rules prescribe recursively how derivable statements can be combined into new derivable statements.

The propositions-as-types interpretation maps logic to typed lambda calculus by interpreting the *propositions* of the logic as the *types* of statements, and the proofs of these propositions as the *terms* inhabiting the types. Under this interpretation there is a direct relation between provability of propositions in the logic and the existence of terms in typed lambda calculus: if a proposition has a proof, there exists a term of the corresponding type. The existence of that term is again decided in relation to a context, which in this case represents all the (logical) assumptions that are in force. Since terms correspond to proofs in this setting, they are sometimes referred to as 'proof objects'. Through the proof objects proofs become first class citizens in type theory, which offers interesting possibilities for the formalization of reasoning *about* proofs inside the system.

Since Howard first gave a description of a propositions-as-types embedding of (first order predicate) logic into (an extension of simply) typed lambda-calculus in 1968[1], a lot of work has been done in this area, motivated by different applications. Originally, the motivation was proof theoretical. Howard wanted to give a formalization of the intuitionistic[2] interpretation of the logical connectives. By representing natural deduction proofs as typed lambda terms, the introduction and elimination rules of the connectives could be phrased as basic operations on lambda terms: lambda abstraction and application. The idea of using typed lambda calculi in the proof theoretical analysis of logics has been taken up and extended by others like Martin-Löf ([Martin-Löf 1984]) and Girard ([Girard et al. 1989]).

[1] His manuscript wasn't published till much later, see [Howard 1980].

[2] The so-called Brouwer-Heyting-Kolmogorov interpretation, see [Troelstra and Van Dalen 1988].

The research project Automath ([Nederpelt et al. 1994]) drew its motivation from mathematics. In 1968 De Bruijn independently defined a propositions-as-types interpretation of logic in typed lambda calculus, with the purpose of using it as a framework for mathematical reasoning in general. By writing mathematics in the language of type theory, it becomes possible to verify mathematical proofs using a computer: the user can present a mathematical proposition (in the form of a type) and a (supposed) proof of this proposition (in the form of a term) to the computer. The computer then checks whether the proof proves the proposition, by checking whether the term is of the given type.

Developments by other research groups have led to systems like Coq ([Dowek et al. 1991]), LEGO ([Luo and Pollack 1992]), and Nuprl ([Constable et al. 1986]), which not only perform proof checking but also give the user some assistance in the construction of proofs.

In computer science, type theory has been used successfully as a framework for the study and development of programming languages. It is possible to view typed lambda calculi as rudimentary but expressive programming languages with the terms functioning as programs and the types as data types (see e.g. [Reynolds 1985]).

In very expressive type theories, such as the Calculus of Constructions ([Coquand and Huet 1988]) and Martin-Löf's Type Theory ([Martin-Löf 1979]), a type can be seen as a complete specification of a program: an intuitionistic (constructive) proof of the proposition $\forall x \in C.pre(x) \supset \exists y \in D.post(x, y)$ contains an algorithm which given an $x \in C$ and a proof that this element satisfies the precondition ($pre(x)$), returns an element $y \in D$ and a proof that the postcondition is satisfied ($post(x, y)$). This algorithm is completely described by the proposition, hence we can identify types with specifications and terms with programs. In this perspective, deriving the existence of a term of a certain type ($A : B$) corresponds to finding a program (A) that satisfies specification (B) (see for instance [Krivine and Parigot 1990]). Usually this 'program' will not be efficient since the term represents the entire proof of the satisfaction of the specification, i.e. it is a mixture of computation and logic. Therefore this application of the propositions-as-types interpretation is usually called 'program extraction', since most of the work in obtaining a feasible program lies in the isolation of the computational content of the proof term (see [Paulin 1989]).

An alternative approach combining these two perspectives on type theory can be found in [Poll 1994]: a program is constructed under the 'data-types-as-types/programs-as-terms' view and its correctness is proved separately under the 'specifications-as-types/programs-as-terms' view. Since both activities take place in the same type system, the typing rules for the program construction can match the proof rules for the correctness proof. In this way, a program and its correctness proof can be developed hand in hand.

Contexts as information states

Currently a new application of type theory is emerging: typed lambda calculi are beginning to be used in knowledge representation. For instance, in [Ahn 1992] a type theoretical approach to user modelling in man-machine communication is proposed. Central to this proposal is the idea that the information state of an agent (animate or inanimate) can be modelled by a type theoretical context. In this view, the assertions that make up an agent's information state are represented as statements, where the type of a statement corresponds to an assertion of the agent and the term inhabiting the type corresponds to the 'justification' or 'evidence' the agent has for this assertion. There are two features of type theoretical contexts that make them suitable for the representation of information states: their 'partiality', and their 'dependency structure'.

In general, the information state of an agent will not contain a complete (or even accurate) description of the world, an agent may be uncertain about some propositions and unaware of others. Contexts match these partial descriptions: the statements in a context represent the propositions for which the agent has evidence, and by means of the derivation rules the logical consequences of these propositions can be deduced. Statements that are not elements of the context or cannot be derived on it, are currently not part of the agents information state. Since the information state of the agent is incomplete, it may 'grow' as he learns more about the world. This growth can be modelled by appending statements representing the new information to the context (cf. [Ranta 1990]). On the extended context, additional statements will be derivable reflecting the consequences of the agents new-found information.

Another feature of contexts is their structure: they represent information states not as a set of formulas, but as a sequence of statements in which each statement may depend on its predecessors. Complicated dependencies can be handled using this structuring. An example of this is the verification of Landau's 'Grundlagen der Analysis' in the Automath language AUT-QE by Van Benthem Jutting ([Van Benthem Jutting 1977]), which shows that the web of dependencies of theorems, lemmas and definitions on previous theorems, lemmas and definitions throughout a mathematical textbook can be treated formally in a typed lambda calculus. The same holds for the anaphoric dependencies in natural language texts that are formalized in Discourse Representation Theory ([Kamp 1981]). Ahn and Kolb ([Ahn and Kolb 1990]) show that the representations of texts generated by DRT can be translated into type theoretical contexts[3].

These two features allow us to represent the development of the information state of an agent by the sequential construction of a type theoretical context. However, type theory also has two basic limitations that have to be dealt with if it is to be used for knowledge representation in a communication setting; these can be labelled as its 'rigidity' and its 'loneliness'.

Type theory is too 'rigid' in the sense that all represented information is of the same kind, i.e. it was designed to deal exclusively with (mathematical) propositions and their proofs. In representing information states we would like to express various degrees of certainty an agent may have about his information, discerning for instance between things the agent 'knows' and things he merely 'believes'.

The 'loneliness' of type theory refers to its mono-logical nature: by the sequential construction of a context we can represent the evolving information state of a single (or 'lonely') agent, whereas an application to communication requires the representation of the (joint) development of the information states of a group of agents.

In logic, these limitations have been 'overcome' by the development of modal logic. The various 'epistemic attitudes' an agent can have towards a proposition (such as knowing it, or believing it) are traditionally dealt with by extending the language with modal operators. Starting from Hintikka's modal logic for one person knowing or believing propositions ([Hintikka 1962]), 'epistemic logics' have been developed that deal with multiple agents and multiple modalities, even with epistemic attitudes of groups of agents like common knowledge.

It is our goal to extend type theory with this approach to intensional reasoning.

[3] A more general discussion of the type theoretical formalization of dependencies in texts can be found in [Ranta 1989]

Contributions of this thesis

In this thesis we present (a class of) type systems in which a propositions-as-types interpretation of (a family of) modal logics is given. The research presented in this monograph is meant to be conducive to the development of type theoretical knowledge representation, but some of its results are also of interest to the disciplines contributing to it: modal logic, proof theory, and type theory.

The proposed 'modal' type systems have a language extended with modal operators, and additional structure in their contexts. By means of the propositions-as-types interpretation, standard systems of epistemic logic can be brought to these type systems. The formal rigour of the embedding guarantees that the intuitions about epistemic reasoning formalized in modal logics are transferred reliably to the type systems, i.e. the modalities in the type system behave exactly like the modal operators in the original modal logic. This means that we can now represent epistemic attitudes of agents towards the assertions in their belief state, as well as the reasoning of agents about their own belief state and those of others.

Reasoning about information states of (other) agents plays an important role in communication. Participants in dialogues often exhibit so-called cooperative behaviour, like not asking your dialogue partner something you already know, or not asking him a question you know he cannot answer. A famous attempt to codify this behaviour are the Gricean maxims ([Grice 1989]). Some of these maxims have already been expressed in terms of the modal operators for knowledge and belief ([Thijsse 1992]). A general formalization of communication based on Gricean principles, will probably involve other modalities expressing intentions ([Appelt 1985], [Beun 1989]) and epistemic attitudes of groups, such as mutual belief ([Jones 1983], [Bunt 1990]). The Fitch-style natural deduction systems for modal logic (described in next section) that underlie the propositions-as-types interpretation allow us to deal with knowledge and belief of multiple agents, and offers prospects for the treatment of intentions and group modalities.

The class of modal type systems to which the interpretation maps modal logics contains very expressive typed lambda calculi, where types may depend on terms. This is also of interest in connection with the formalization of dialogues, since it allows us to combine the dynamic (DRT-)representation of natural language (for which this expressivity is needed) with the modal reasoning used for modelling cooperative behaviour.

The interest our research may have for each of the disciplines that fostered it, can be indicated as follows.

Proof theory: The propositions-as-types interpretation establishes an isomorphism between modal natural deduction proofs and type theoretical terms. This offers the opportunity to formalize reductions on modal proofs as reduction rules in type theory. A number of these reductions are defined in our framework and are proven to be well-behaved.

Modal logic: The primary contribution to this field consists in the straightforward generalization of Fitch-style modal deduction to multi-agent multi-modal systems. However, in a more general perspective our work ties in with the current move towards formulating logics (for AI and linguistics) in a binary format, as propagated by Gabbay's Labelled Deductive Systems research program ([Gabbay 1993]). In [Van Benthem 1991a] this approach is proposed for epistemic logic, with the aim of incorporating justifications of knowledge into the logic as first-class citizens. The terms in our modal type systems indicate what the explicit calculus of justifications induced by traditional epistemic logics looks like.

Type theory: Although the modal type systems are a generalization of the Pure Type Systems (described below), there currently seem to be no intrinsic type theoretical reasons for studying them. However, there may be applications of type theory (other than in knowledge representation) which could benefit from the use of modalities along the lines of our work.

Frameworks

As the reader will have gathered from the brief overview above, there are many systems of typed lambda calculus around, formulated with different objectives in mind. However, in recent years some unification and standardization has been achieved. Barendregt noticed that many of the existing systems of (explicitly) typed lambda calculus could be uniformly represented in a framework which is parametrized with respect to the derivation rules ([Barendregt 1992]). This insight eventually led to the format of Pure Type Systems (PTSs); a general description of a large class of typed lambda calculi, for which most of the desirable meta theoretical properties can be proved generically. This attractive feature has rapidly gained PTSs a central position in research in type theorety.

A more specific property that makes PTSs such a suitable type theoretical framework for our investigations, is their well-studied connection with (non-modal) logic. The work of Geuvers ([Geuvers 1993]) gives a comprehensive and detailed account of the propositions-as-types interpretation of first and higher order propositional and predicate logics. The 'modal' typed lambda calculi presented in this thesis are an extension of the PTSs, unsurprisingly called 'Modal Pure Type Systems' (MPTSs). In studying their properties and connection to modal logic we fruitfully and gratefully use theory and techniques developed in the PTS-framework.

Finding a suitable framework for the natural deduction formulation of modal logics is a problem. Despite the spectacular advances in the model theory of modal logic, its proof theory has remained underdeveloped. Recently, interest in modal proof theory has rekindled and new approaches have been developed, e.g. to modal sequent calculi ([Wansing 1992],[Martini and Masini 1993]). However, to our knowledge this has not lead to 'Prawitz-style' natural deduction systems or 'Gentzen-style' sequent calculi for the modal logics we are interested in in this thesis. In lieu of these traditional tools, we use lineair, 'Fitch-style', natural deduction systems as a basis for the propositions-as-types interpretation.

Starting from [Fitch 1952] which gives a deduction system for a single modal logic (S4), this approach has evolved to a point where a modular presentation can be given of a reasonable number of the modal logics (above K) that are most common in literature (see [Fitting 1983]). This thesis shows that, despite its limited scope, Fitch-style deduction is a well-behaved and intuitive framework for the logics we are interested in. Moreover, it is flexible enough to allow generalization to logics with multiple agents and multiple modalities.

The contents of this thesis

We start with an introductory chapter in which we provide a natural deduction formulation of a family of normal modal logics, and present the Modal Pure Type Systems. Chapter 2 gives a detailed account of the propositions-as-types interpretation of the modal logics in MPTSs.

We define mappings from natural deduction proofs to typed lambda terms and vice versa, and prove soundness for these mappings as well as some invariance results for their composition. In addition, reduction rules on MPTS-terms are proposed that correspond to proof reductions on the modal natural deductions. In chapter 3 we show that the MPTSs are a well-behaved extension of the PTSs, by proving that the desirable meta theoretical properties of the PTSs are preserved.

The fourth chapter discusses how both the modal natural deduction systems and the MPTSs can be generalized to deal with multiple agents and multiple modalities. This discussion is guided by Wiebe van der Hoek's work on the system KB_{CD} of [Kraus and Lehmann 1986] (see [Van der Hoek 1992]). The subject of chapter 5 is another strengthening: changing the logic underlying the modal systems to predicate logic. Section 5.1 analyzes the consequences of this move for the interpretation described in chapter 2. In the second part of the chapter, we re-examine familiar problems of modal predicate logic in the setting of MPTSs. In chapter 6 we indicate how the MPTSs could be put to work in the formalization of communication. The final chapter contains some concluding remarks and directions for future research.

Readers who are not interested in the technical details of the interpretation and the meta theory of modal type systems, are advised to take a short cut through the thesis by skipping chapters 2 and 3. The introduction (chapter 1) is intended to provide them with sufficient understanding of the MPTSs, and feel for the propositions-as-types interpretation to make their way through chapters 4-7 (with exception of sections 4.2.4 and 5.1 which presuppose familiarity with chapter 2).

In this thesis we combine two well-established sports: modal logic and typed lambda-calculus. Since few people play in both fields, we have tried to keep this thesis self-contained enough to be read by modal logicians as well as type theoreticians. A drawback of this approach is that things can become too self-contained in sections dealing with one's native discipline. We kindly ask the reader to view this as a tribute to his knowledge, rather than a presumption of his ignorance.

Acknowledgements

I want to thank my promotors prof.dr. J.C.M. Baeten and prof.dr. H.C. Bunt for their willingness to immerse themselves in the subject of this thesis. Their comments, as well as those of the other members of the manuscript committee, prof.dr. H.P. Barendregt and prof.dr. J.F.A.K. van Benthem, were very helpful.

Thanks are also due to the SOBU, in particular drs. A. Pillot and drs. M. Wagemans, for providing me with the opportunity to work in such an interdisciplinary project as DenK. The people working in DenK project provided a stimulating environment; discussing your work with the likes of René Ahn (ITK/KUB), Jan Jaspars and Robbert-Jan Beun (IPO) really helps to develop your ideas.

A lot of contacts outside of the DenK-project have been of importance to me, of which I would like to mention Heinrich Wansing, Elias Thijsse, the participants of the Seminar on Intensional Logic and its' organizer Maarten de Rijke, and the participants and organizers of the λ Intercity Seminar (although I still feel that lectures in these seminars should not have taken place simultaneously).

Closer to home, I want to thank my past and present colleagues at the group 'Formele

Methoden' (previously 'Theoretische Informatica') and the AIOs and OIOs of the discrete mathematics group, especially Patrick Bours and René Struik, for providing me with such lively and pleasant surroundings.

Of the people who helped my find my way in type theory I would like to thank: Bert van Benthem Jutting for teaching me that hardly anything is obvious in this field, Erik Poll for helping me escape from some vicious circles in the meta theoretical proofs, and Herman Geuvers for writing a thesis on logic and type systems just when I needed it, and for subsequently becoming my room-mate to explain the parts I didn't understand.

A lot of thanks are due to my paranymphs Robert Cuperus and Niels Lous for helping me in preparing this thesis, even up to the point where they litteraly became my left and right hand while I was unable to type.

Special thanks are due to my supervisor Rob Nederpelt for helping me find my way in scientific research. His patient and honest guidance in theoretical as well as practical matters has been of great value to me.

Finally I want to thank my parents for their continual support and encouragement.

When I started writing this thesis I expected to get by 'with a little help from my friends', I didn't know that it would take the Joe Cocker-version. Thanks again!

Chapter 1

Introduction

This chapter introduces the two formalisms that will be starring in this thesis. In the first section we give a natural deduction formulation of a family of modal logics. The second section presents the Modal Pure Type Systems in which these logics are to be interpreted. To give the reader a preliminary idea of the interpretation, we begin this presentation with a description of the Pure Type Systems (on which the MPTSs are based) and their relation to propositional and predicate logic.

1.1. Modal natural deduction

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 lineair proofs. For modal logic the vast majority of systems in the literature is linear[1]. 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. His original system works only for one particular modal logic ($S4$), but in [Fitch 1966a] the idea is successfully applied to other logics. Using some further extensions as given in [Siemens 1977], Fitting ([Fitting 1993]) is able to give a modular presentation of Fitch-style deduction systems for the 'normal' modal logics that are most common in literature. Before explaining the deduction rules, a short introduction to these modal logics and 'normality' will be given.

1.1.1. Normal modal logic

Technically, modal logic is an extension of propositional or predicate logic with the operators '$\Box$', expressing necessity, and '$\Diamond$', expressing possibility. Given a propositional language consisting of proposition letters $A_1, A_2, \ldots, B, \ldots$, constants $\top$ and $\bot$, and connectives $\neg, \wedge, \vee, \supset, \leftrightarrow$; $\Box\varphi$ ('necessarily φ') and $\Diamond\psi$ ('possibly ψ') are well-formed formulas if φ and ψ are well-formed formulas.

A modal logic is considered 'normal' if:

- its modal operators are related by the definition: $\Diamond\varphi \leftrightarrow \neg\Box\neg\varphi$

[1] An exception being a paper by Fitch! ([Fitch 1966b]).

- it is closed under the following rule:

$$\textit{Normality}\ \ \frac{(\varphi_1 \wedge ... \wedge \varphi_n) \supset \psi}{(\Box\varphi_1 \wedge ... \wedge \Box\varphi_n) \supset \Box\psi}\ (n \geq 1)$$

The smallest normal modal logic has just this rule and definition. It is the well-known logic K, which can alternatively be characterized as the set of propositions derivable by means of:

- all propositional tautologies
- axiom: $\Box(\varphi \supset \psi) \supset (\Box\varphi \supset \Box\psi)$
- rules:

$$\textit{Modus Ponens}\ \ \frac{\varphi \quad \varphi \supset \psi}{\psi}$$

Necessitation: if φ *is a thesis, then* $\Box\varphi$ *is a thesis*

(where a *thesis* is a well-formed formula that is an axiom or a theorem of the logic).

K can be strengthened by adding further intuitive properties of necessity in the form of axioms. Throughout this thesis we will consider normal systems resulting from the extension of K with one or more of these axiom(schema)s:

D : $\Box\varphi \supset \Diamond\varphi$

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

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

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

B : $\varphi \supset \Box\Diamond\varphi$

The following convention will be used in referring to the normal extensions of K:

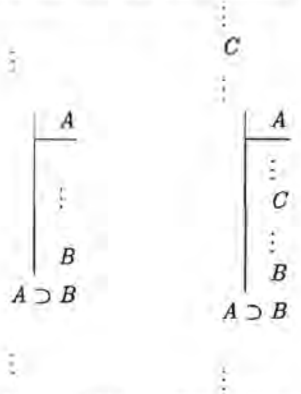
> $KS_1\ ...\ S_n$ = the smallest normal system of modal logic containing (every instance of) the schemas $S_1\ ...\ S_n$.

Combining K with these axioms yields 14 different normal modal logics[2] above K, which are well-behaved and common in literature. These standard logics are a firm point of departure for our type theoretical interpretation, and also of a more specific interest for this thesis: the axioms D, T, 4, and 5 are important principles in epistemic and doxastic logic (where '$\Box$' is interpreted as 'it is known that', and 'it is believed that' respectively), and the B-axiom is of technical interest in modal predicate logic.

[2]See [Chellas 1980].

1.1.2. Natural deduction rules for K

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.

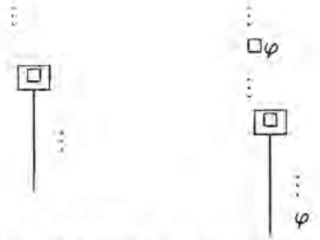
To extend his deduction system to modal logic, Fitch added a new kind of subordinate proof, the *strict* subordinate proof. It differs from 'ordinary' subordinate proofs in two respects:

- 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 '$\Box$' on top of the vertical line, which indicates the *modal interval*.

For the logic K reiteration is restricted to formulas of the general form $\Box\varphi$: 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:

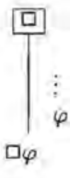


A strict subordinate proof K-import

K-import: φ may occur in a strict subordinate proof if $\Box\varphi$ 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 ($\Box\varphi$). 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.

In terms of possible worlds the procedures for import and export can be understood in the following way: if we take a main proof to be the world in which we try to establish the truth of a modal formula, a strict subordinate proof corresponds to an arbitrary accessible world. In such a world we only know the truth of the propositions (φ) that were necessary ($\Box\varphi$) in the original world. In this view, starting a strict subordinate proof amounts to continuing the proof in an arbitrary accessible world. Every proposition (ψ) that can be derived without hypotheses in such a world could have been derived in any accessible world, hence it can be considered necessary in the original world ($\Box\psi$). In this way conclusions obtained in the accessible world can be brought back (exported) to the world where the proof was started, and the proof can be resumed there.

To illustrate the use of these rules we prove an instance of the K-axiomschema.

1.	$\Box(A \supset B)$	
2.	$\Box A$	
3.	$\Box(A \supset B)$	(reiteration, 1)
	$\boxed{\Box}$	
4.	$A \supset B$	(K-import 3)
5.	A	(K-import 2)
6.	B	($\supset$-elim 4,5)
7.	$\Box B$	(K-export 6)
8.	$\Box A \supset \Box B$	($\supset$-intro 2-7)
9.	$\Box(A \supset B) \supset (\Box A \supset \Box B)$	($\supset$-intro 1-8)

The formula $\Box(A \supset B)$ has to be reiterated (line 3) before it can be imported. This is due to conditions on the application of K-import that will be specified in forthcoming definitions.

Adding the K-import rule and the K-export rule to a Fitch-style deduction system for classical propositional logic yields a deduction system for K.

1.1.3. $\Box PROP_{fitch}$

We now give a formal definition of $\Box PROP_{fitch}$, a Fitch-style deduction system for the modal logic K. 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 in later chapters, where 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 $\Box PROP_{fitch}$ 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.

1.1.1. DEFINITION. **Proof figure**
A *proof figure* $\mathbf{D}$ is a mathematical structure consisting of:

1 an interval $D = [1, n]$, where $D \subset IN$,

2 a function $F : D \to PROP$, 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 disjunct subcollections $\mathbf{H}$ and $\mathbf{M}$:

  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 $[k, l]$.

  M the *modal intervals* of the proof figure. D may not be an element of $\mathbf{M}$.
  If $[k, l] \in \mathbf{M}$ then the formula F_k is not a hypothesis of $[k, l]$.

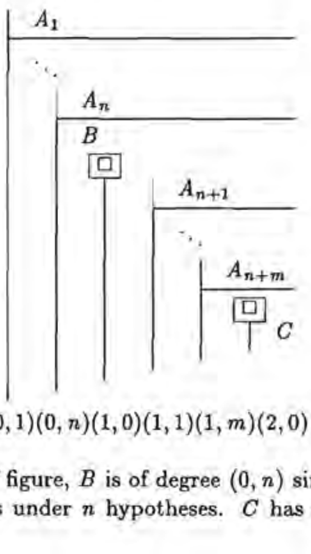
In Fitch-style deduction for non-modal propositional logic **I**=**H**; every subinterval is a hypothesis interval introduced by assuming the topmost formula of that interval. The presence of a modal interval in a proof figure does not require an assumption and hence the topmost formula of such an interval 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 ('□') on top of its vertical line.

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

1.1.2. 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 *preceeds* 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 $i \in J \subset I$, than 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 (as in the figure below), 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 deepest modal interval of which F_i is an element.



$(0,0)$ $(0,1)(0,n)(1,0)(1,1)(1,m)(2,0)$ (degree)

In this schematic modal proof figure, B is of degree $(0,n)$ since there are no modal intervals to the left of it, and it occurs under n hypotheses. C has degree $(2,0)$ since it *lies in* the second modal interval. The hypotheses $A_{n+1}, \ldots, A_{n+m}$ have degree $(1, i)$ with $(1 \leq i \leq m)$, since they lie in the i-th hypothesis-interval to the right of the first modal interval.

1.1.3. DEFINITION. **Degree**

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

The natural deduction rules of $\Box PROP_{fitch}$ 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.

1.1.4. DEFINITION. **Deduction rules**

$\vee$-*intro*

A B
⋮   ⋮
$A \vee B$ $A \vee B$

$\vee$-*elim*

$A \vee B$
⋮
| A
| ⋮
| C
⋮
| B
| ⋮
| C
⋮
C

$\neg$-*intro*

| A
| ⋮
| B
| ⋮
| $\neg B$
$\neg A$

$\neg$-*elim*

$\neg\neg A$
⋮
A

⊃-intro

A
$\vdots$
B
$A \supset B$

⊃-elim

$\vdots$
$A \supset B$
$\vdots$
A
$\vdots$
B

∧-intro

A
$\vdots$
B
$\vdots$
$A \wedge B$

∧-elim

$A \wedge B$
$\vdots$
A

$A \wedge B$
$\vdots$
B

↔-intro

$A \supset B$
$\vdots$
$B \supset A$
$\vdots$
$A \leftrightarrow B$

↔-elim

$A \leftrightarrow B$
$\vdots$
$A \supset B$

$A \leftrightarrow B$
$\vdots$
$B \supset A$

reiteration

$\vdots$
A
$\vdots$
$\ddots$
$\vdots$
A

K import

$\vdots$
$\Box A$
$\vdots$
$\boxed{\Box}$
$\vdots$
A

K export

$\boxed{\Box}$
$\vdots$
A
$\Box A$

1.1.5. DEFINITION. **Application of deduction rules**
Given a proof figure **D**, with interval $D = [1, n]$, formulas $F_1, \ldots, F_n$ and intervals **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 in the proof figure, and one of the following conditions is met:

1 $R \in \{\vee\text{-}intro, \neg\text{-}elim, \supset\text{-}elim, \wedge\text{-}intro, \wedge\text{-}elim, (\leftrightarrow\text{-}intro, \leftrightarrow\text{-}elim)\}$.
In this case the premisses and the conclusion E all lie in the *same interval*. The order in the which the premisses appear may differ from the one given in the table.

2 $R = \neg - intro$.
there has to be a hypothesis-interval $[k,l] \in$**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 $R = \supset\text{-}intro$.
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 $R = \vee\text{-}elim$.
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.

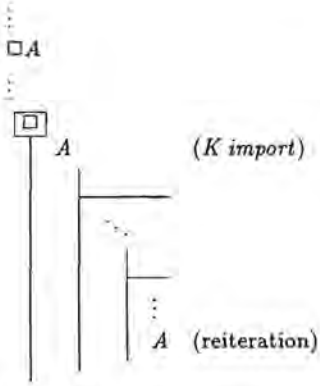
5 $R = reiteration$.
If the premiss A lies in the interval $I \in \mathbf{I} \cup \{D\}$ and the conclusion $E = A$ lies in the interval $J \in \mathbf{I} \cup \{D\}$, then it has to be the case that $(J \subseteq I) \wedge \neg\exists K \in \mathbf{M}.(J \subset K \subseteq I)$. Or, in terms of modal 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 $\Box A$ lies in interval $I \in \mathbf{I}$ and the conclusion $E = A$ lies in the interval $J \in \mathbf{M}$, 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 $I \in \mathbf{M}$ and the conclusion $E = \Box A$ lies in the interval $J \in$**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.

In the modal system *reiteration* is defined in such a way that (arbitrary) formulas may only be repeated in subordinate proofs that have the same modal depth as the interval in which the original occurrence lies. Note that, although *K import* is limited to the modal interval immediately to the right of the hypothesis-interval, combining *K import* and *reiteration* makes it possible to import the formula φ 'over' any number of assumptions (hypothesis intervals) lying inside the modal subordinate proof:



1.1.6. Definition. Derivation without hypotheses
A *derivation* of a formula C is a proof figure $\mathbf{D}$ with interval $D = [1, n]$ and formulas $F_1, \ldots F_n$, 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.

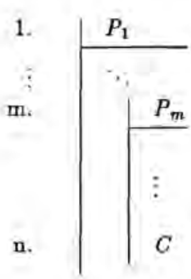
1.1.7. Definition. Derivation with hypotheses
A *derivation* of a formula C from the formulas $P_1, \ldots, P_m (m \geq 1)$ is a proof figure $\mathbf{D}$ with interval $D = [1, n]$ $(n > m)$ and formulas $F_1, \ldots F_n$, that satisfies the following conditions:

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 $P_1, \ldots, P_m$ 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:



The actual proof of C (denoted by the vertical dots) is then a derivation in the hypothesis interval of P_m.

1.1.8. 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 $P_1, \ldots P_m$ if there exists a derivation of C from $P_1 \ldots P_m$, written as $P_1, \ldots, P_m \vdash C$.

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 $P_1, \ldots, P_m \in \Gamma$ such that $P_1, \ldots, P_m \vdash C$. This is written: $\Gamma \vdash C$. If $\Gamma = \emptyset$, $\vdash C$.

1.1.4. Modal deduction for extensions of K

We distinguish two ways in which a natural deduction system for K can be extended to accommodate stronger normal modal logics: extension by axioms and extension by rules. In the first case, added axioms determine the modal strength of the deduction system. In the second case, additional import- and export-rules, governing the exchange of formulas between proofs and strict subordinate proofs, determine the strength of the system. After the presentation of these two approaches, it will be shown that they are equivalent for the systems we are concerned with.

Extension by axioms

In this approach, modal axioms are added to the proofs as some sort of tacit assumptions; they are formulas that may be written at any stage in any proof without further justification. In this way conclusions not available in K can be reached by proving the antecedent of an axiom with the rules of K and then, after writing that axiom in the proof, deriving the consequent of the axiom by *Modus Ponens*. Although any modal axiom could be added in this way, we restrict ourselves to (combinations of) the following axioms:

$D : \Box A \supset \Diamond A$

$T : \Box A \supset A$

$4 : \Box A \supset \Box\Box A$

$5 : \neg\Box A \supset \Box\neg\Box A$

$B : A \supset \Box\Diamond A$

As an example, we show how A can be derived from $\Box A$ after extending K with the T-axiom.

$$\begin{array}{ll} \vdots & \\ \Box A & \\ \Box A \supset A & (T\text{-axiom}) \\ A & \\ \vdots & \end{array}$$

Extension by rules

The deduction system for K can also be strengthened by adding import and export rules in order to:

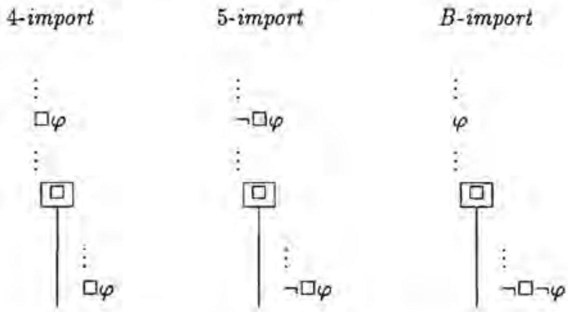
> Increase the number of propositions derivable in strict subordinate proofs by providing them with more 'input'. This can be done by allowing more (kinds of) formulas to be reiterated in these proofs: add import rules.

> Make better use in the main proof of the propositions derived in the subordinate proofs (their 'output'). This can be done by making the conclusions of the strict subordinate proofs available to the main proof in more than one (the necessitated) form: add export rules.

For all the axioms we are concerned with, a single import or export rule can be given that, when added to the rules for K, strengthens the deduction system in the same way as the axiom: for the axioms 4, 5, and B an extra *import rule* is needed, for the axioms D and T an extra *export rule*. First each of the extra rules will be given along with its structural form. Then we will show how these rules can be derived in the presence of 'their' axioms, and how the axioms can be obtained from the rules.

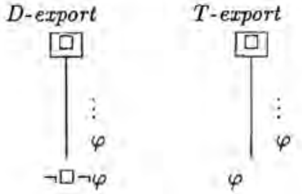
The import rule corresponding to the 4-axiom allows the reiteration of formulas of the form $\Box\varphi$ in a strict subordinate proof without changing their form. Similarly, the rule corresponding to the 5-axiom allows the 'verbatim' reiteration of formulas of the form $\neg\Box\varphi$. *Any formula* φ may be reiterated as $\neg\Box\neg\varphi$ in a strict subordinate proof, using the rule for the B-axiom.

1.1.9. DEFINITION. **Import rules**



The export rule corresponding to the D-axiom permits a formula φ to be brought back to the main proof in the form $\neg\Box\neg\varphi$. The export rule for the T-axiom allows any formula from the strict subordinate proof to be brought back to the main proof without changes.

1.1.10. Definition. Export rules



To complete the formal definition of these rules in $\Box PROP_{fitch}$ we complement the figures above by the following definition.

1.1.11. Definition. Application of deduction rules
Given a proof figure **D**, with interval $D = [1, n]$, formulas $F_1, \ldots, F_n$ and intervals **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 in the proof figure, and one of the following conditions is met:

8 $R = 4\ import$.
If the premiss $\Box A$ lies in interval $I \in \mathbf{I}$ and the conclusion $E = \Box A$ lies in the interval $J \in \mathbf{M}$, then it has to be the case that *the interval* J lies in *the interval* I.

9 $R = 5\ import$.
If the premiss $\neg\Box A$ lies in interval $I \in \mathbf{I}$ and the conclusion $E = \neg\Box A$ lies in the interval $J \in \mathbf{M}$, 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 $I \in \mathbf{I}$ and the conclusion $E = \neg\Box\neg A$ lies in the interval $J \in \mathbf{M}$, then it has to be the case that *the interval* J lies in *the interval* I.

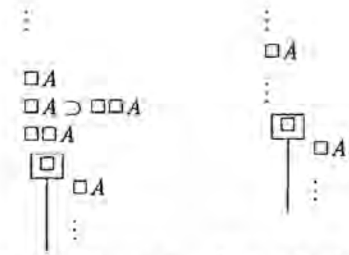
11 $R = D\ export$.
If the premiss A lies in interval $I \in \mathbf{M}$ and the conclusion $E = \neg\Box\neg A$ lies in the interval $J \in \mathbf{I}$, then it has to be the case that *the interval* I lies *in the interval* J.

12 $R = T\ export$.
If the premiss A lies in interval $I \in \mathbf{M}$ 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.

From axioms to rules

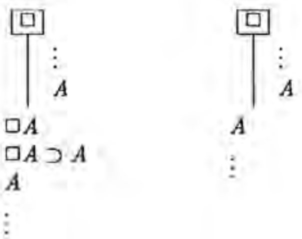
In the natural deduction system for K extended with the 4, 5, B, D, and T axioms, the import and export rules given above are derivable.

The 4-, 5- and B-import rules can be shown to be derived rules using the corresponding axiom and the K-import rule. We show this for 4-import: a formula of the form stipulated by the rule ($\Box A$) is used to derive the consequens of the axiom ($\Box\Box A$) to which the K-import rule is then applied. The result of this procedure is the same as that of applying the extra import rule directly to the formula of the required form.



4-axiom & K-import   4-import rule

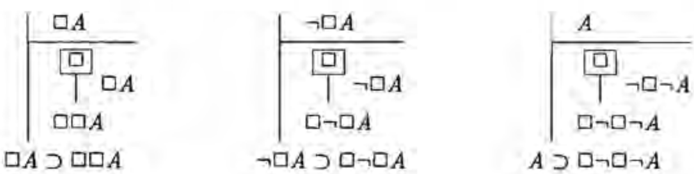
The D- and T-export rules can be shown to be derived rules using the corresponding axiom and the K-export rule; after the application of K-export to a conclusion (A) of the strict subordinate proof the resulting formula ($\Box A$) is used to derive the consequens of the axiom ($\neg\Box\neg A$, or A). The result of this procedure is the same as that of a direct application of the extra export rule to the conclusion of the strict subordinate proof. For example:



T-axiom & K-export T-export rule

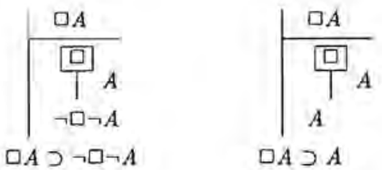
From rules to axioms

The 4, 5, and B-axioms can be derived from the new rules as follows. First assume the axiom's antecedent, then apply the import rule corresponding to the axiom, immediately followed by K-export.



4-axiom   5-axiom B-axiom

Similarly, the D and T-axioms can be derived by subsequent application of K-import and their corresponding export rules.



D-axiom *T*-axiom

1.1.5. Deduction systems

In this section we first give natural deduction systems for the extensions of K with combinations of 4, 5, B, D, and T, and compare these to Fitch-style systems in the literature. Then we briefly discuss the general prospects of 'extension by rules' versus 'extensions by axioms'.

Given the equivalence of axioms and rules, a natural deduction system for a normal extension $KS_1 \ldots S_n$ of K can be found by adding to a system for classical propositional deduction the rules for K-import, K-export and:

- all instances of the axiomschemas $S_1 \ldots S_n$ (extension by axioms), **or**
- the import- or export-rules corresponding to $S_1 \ldots S_n$ (extension by rules).

Of course 'mixed systems' with both added axioms and rules are also possible. All of the Fitch-style systems in literature ([Fitch 1952], [Siemens 1977], [Fitting 1983]) are mixed in this sense: 4, 5 and B are dealt with by import rules ('modifying strict reiteration'), D and T using axioms. Instead of allowing the D and T axiom to be written anywhere in a proof, 'rules' are used as shorthand for deriving the consequent of an axiom by modus ponens: a 'D-rule' that allows the derivation of $\neg\Box\neg A$ from $\Box A$, and a 'T-rule' for the direct inference of A from $\Box A$. A reason for this mixed approach may be that originally only normal systems at least as strong as KT were considered, where '$\Box$' can be treated on a par with the logical connectives; it has an introduction rule (K-export) and an elimination rule (the 'T-rule').

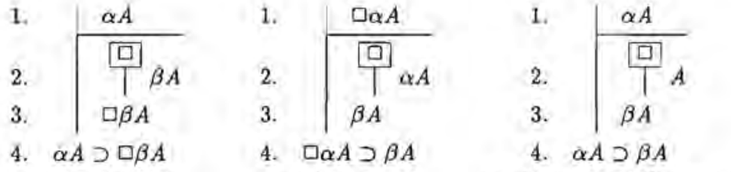
By varying both import and export rules we put more emphasis on the structural role of the modalities: a modal proof is conceived of as a group of propositional deductions between which formulas may be exchanged. The import and export-rules governing this exchange determine the modal strength of the system. This 'separation of concerns' will prove to be of interest in the interpretation of the extension-by-rules-systems in typed λ-calculus.

Unfortunately, little seems to be known of the general scope of 'rules' in Fitch-style deduction. Hawthorn ([Hawthorn 1990]), commenting on [Fitch 1966a], is not very hopeful for logics other than the ones treated above:'These systems have all the virtues of a good natural deduction system: they are easy to work in and provide a feel for the deductive structure of the respective logics. The disadvantage is that these virtues begin to disappear as soon as one tries to extend their methods to other logics. The method of varying the rule of strict reiteration works only for axioms of the form $\alpha A \supset \Box\beta A$, where α and β are modalities, and the number of axioms that can plausibly thought of as embodying a $\Box E$ rule is extremely small. Very soon, one has to admit defeat and just add axioms, as Segerberg[3] advocated.'

The left figure below shows the general format for proving axioms of the form $\alpha A \supset \Box\beta A$ by varying the import rule: the idea is to add an import rule that imports a formula of the

[3]See [Bull and Segerberg 1984].

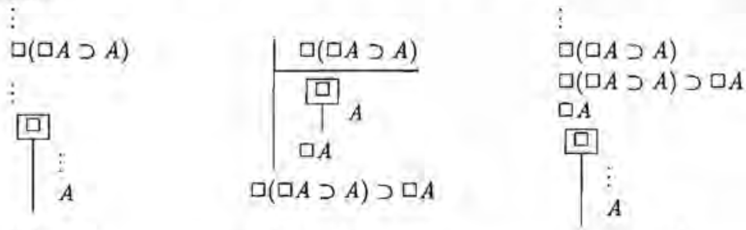
form αA as βA in the modal subordinate proof, K-export will then yield $\Box\beta A$. Besides 4, 5, and B, this scheme gives us rules for axioms like $\Diamond\Box A \supset \Box\Diamond A$. This axiom is characteristic of a well-known extension of $KT4$, $S4.2$.



varying *import* varying *export* varying *import & export*

By the same reasoning, varying the export rule gives us rules for axioms of the general form $\Box\alpha A \supset \beta A$ (middle figure above): K-import is used to import $\Box\alpha A$ as αA in the modal subordinate proof, an additional export rule then exports αA as βA. Other than D or T, this suggest rules for an axiom like $\Box\Diamond A \supset \Diamond\Box A$, the characteristic axiom of another well-known extension of $KT4$ ($S4.1$).

To get a grip on the expressibility of the import- and export- rules, we need to answer the question which of the many possible variations on this theme are reasonable and meaningful rules. The rightmost figure above shows that any axiom $\alpha A \supset \beta A$ is trivially derivable when we are allowed to add an import- export-rule pair for it. Hence it seems reasonable to restrict ourselves to axioms that can be derived adding only an import rule or only an export rule. But even then questions remain, like does this always have to be derivability with respect to the basic logic K? Another question is whether we allow import and export rules to change the 'matrix' of the formula rather than just the modalities, like in the following import rule for Löb's axiom:



Löb-import rule to axiom axiom to rule

Regardless of the general prospects of Fitch-style modal deduction, the system $\Box PROP_{fitch}$ proves to be a very well-behaved and intuitive framework for the normal modal logics we are interested in in this thesis. In later chapters its flexibility will be exploited by extending it to accommodate multi-agent and multi-modal logics.

1.2. Modal Pure Type Systems

In this section we propose a set of systems of typed λ-calculi in which several normal modal propositional and predicate logics can be interpreted. These systems are obtained by extending the format of the Pure Type Systems (PTSs) popularized by Barendregt ([Barendregt 1991], [Barendregt 1992]). Under the so-called 'propositions-as-types'interpretation, Fitch-style deduction for propositional and predicate logic can already be accommodated in these typed λ-calculi: there are type theoretical analogons of 'proofs' and 'subordinate proofs'. The idea behind the proposed extension is to add an analogon of 'strict subordinate proof' to the PTSs and then see if the modal natural deduction rules have a counterpart in such a framework.

The structure of this section mirrors that of section 1.2; first an introduction to PTSs and the propositions-as-types interpretation of propositional logic will be given, then the extra rules for the smallest modal system (K) will be explained followed by the formal definition of the 'Modal Pure Type Systems' and two ways to extend these type theoretical systems: an extension by rules, and an extension by axioms.

1.2.1. Pure Type Systems

Since the beginning of typed lambda calculus[4], many different systems have been proposed with different applications in mind. They can be divided into systems with 'implicit' and 'explicit' typing. In *implicit* or *Curry-typing* one tries to find a (most general) suitable type for a given untyped lambda-term. In the more common *explicit* or *Church-typing*, type information is inserted in the term: variables are introduced along with a type. The type of a lambda-term can then be built during its construction from the types of the variables, according to certain derivation rules.

Barendregt noticed that many of the existing systems of lambda calculus with explicit typing could be uniformly represented in a framework which is parametrized with respect to these derivation rules ([Barendregt 1991]). This insight eventually led to the format of Pure Type Systems; a general description of a large class of typed lambda calculi, providing possiblities for generic proofs of meta theoretical properties.

PTSs are formal systems for deriving judgements of the form $\Gamma \vdash M : A$, meaning that type 'A' can be assigned to term 'M' in context 'Γ' (also pronounced as: 'M is of type A', or 'M is in A is derivable in context Γ'). Both M and A are elements of the set of so-called *pseudoterms*, a set of expressions from which the derivation rules select the ones that are fit in a judgement. Since PTSs have explicit typing, the typing of a term M with a type A can only be done relative to a typing of the free variables that occur in M and A. This information is recorded in the context Γ, which is a finite sequence of so-called declarations, statements of the form $x : B$ where x is a variable and B is a pseudoterm.

We start the definition of Pure Type Systems by defining the set of pseudoterms $\mathcal{T}$ from a base set $\mathcal{S}$ of constants called 'sorts'.

1.2.1. DEFINITION. **Pseudoterms**
For $\mathcal{S}$ some set, the set $\mathcal{T}$ of pseudoterms over $\mathcal{S}$ is

$$\mathcal{T} ::= \mathcal{S} \mid Var \mid (\Pi\, Var : \mathcal{T}.\mathcal{T}) \mid (\lambda\, Var : \mathcal{T}.\mathcal{T}) \mid (\mathcal{T}\mathcal{T})$$

where *Var* is a countable set of variables.

Both Π and λ bind variables and hence we have the usual notion of *free variable* and *bound variable*. Like in untyped λ-calculus we have β-reduction[5] on $\mathcal{T}$: $(\lambda x : A.M)P \rightarrow_\beta M[P/x]$, where $M[P/x]$ denotes the substitution of P for all free xs in M (avoiding capturing of free variables by renaming). The transitive reflexive closure of $\rightarrow_\beta$ is denoted by $\twoheadrightarrow_\beta$ and the

[4]Usually marked with [Curry 1934] and [Church 1940].

[5]η-reduction could also be added, see [Geu 93], but will not be considered here.

transitive symmetric closure of $\twoheadrightarrow_\beta$ by $=_\beta$. Syntactical identity of pseudoterms A and B is denoted by $A \equiv B$.

1.2.2. DEFINITION. **Contexts**

(i) A *statement* is an expression of the form $A : B$, with $A, B \in \mathcal{T}$. The pseudoterm A is called the *subject* of the statement $A : B$. A *declaration* is a statement 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: $x_1 : A_1, \ldots, x_n : A_n$. The empty context is denoted by ε. If $\Gamma = x_1 : A_1, \ldots, x_n : A_n$ then $\Gamma, x : B = x_1 : A_1, \ldots, x_n : A_n, x : B$.

1.2.3. DEFINITION. **Pure Type Systems**
A *Pure Type System with β-conversion*, PTS_β, is given by a set $\mathcal{S}$ of *sorts* , a set $\mathcal{A} \subset \mathcal{S} \times \mathcal{S}$ of *axioms*, and a set $\mathcal{R} \subset \mathcal{S} \times \mathcal{S} \times \mathcal{S}$ of *rules*. The PTS that is given by $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{R}$ is denoted by $\lambda_\beta(\mathcal{S}, \mathcal{A}, \mathcal{R})$ and is the typed λ-calculus with the following deduction rules:

$$(axiom)\ \ \varepsilon \vdash s_1 : s_2 \quad \text{if } s_1 : s_2 \in \mathcal{A}$$

$$(start)\ \ \frac{\Gamma \vdash A : s}{\Gamma,\ x : A \vdash x : A}$$

$$(weakening)\ \ \frac{\Gamma \vdash A : B \quad \Gamma \vdash C : s}{\Gamma,\ x : C \vdash A : B}$$

$$(product)\ \ \frac{\Gamma \vdash A : s_1 \quad \Gamma,\ x : A \vdash B : s_2}{\Gamma \vdash (\Pi x : A.B) : s_3} \quad \text{if } (s_1, s_2, s_3) \in \mathcal{R}$$

$$(application)\ \ \frac{\Gamma \vdash F : (\Pi x : A.B) \quad \Gamma \vdash a : A}{\Gamma \vdash Fa : B[a/x]}$$

$$(abstraction)\ \ \frac{\Gamma,\ x : A \vdash b : B \quad \Gamma \vdash (\Pi x : A.B) : s}{\Gamma \vdash (\lambda x : A.b) : (\Pi x : A.B)}$$

$$(conversion)\ \ \frac{\Gamma \vdash A : B \quad \Gamma \vdash B' : s \quad B =_\beta B'}{\Gamma \vdash A : B'}$$

where s ranges over $\mathcal{S}$, the set of sorts, x ranges over variables, and it is assumed that in the rules (*start*) and (*weakening*) the newly declared variable x is always fresh, that is it has not yet been declared in Γ. These rules axiomatize the notion $\Gamma \vdash A : B$ stating that $A : B$ can be derived from context Γ; in that case A and B are called *legal expressions* and Γ is a *legal context*.

All rules except *product* hold uniformly for all PTSs; given a set of sorts $\mathcal{S}$ and axioms $\mathcal{A}$, systems differ only in the instantiations of the *product* rule that are allowed by $\mathcal{R}$. Rather than explaining the PTS-rules in this general format, we focus on a small number of systems which are of importance for the interpretation of logic. These are the PTSs on which the work in this thesis is based, and it is convenient to introduce the PTS-rules in the setting in which we will be using them.

There are several ways in which logic can be coded into typed λ-calculus. One of the more direct methods is to simply interpret the *propositions* of logic as *types* in typed λ-calculus. The *proofs* of propositions are then coded as *terms* (called proof objects) of the type corresponding to those propositions. Under this so-called 'propositions-as-types' interpretation[6] there is a direct relation between provability in logic and the existence of terms in typed λ-calculus: if a proposition has a proof there exists a term of the corresponding type (this type is then said to be 'inhabited' by that term).

The following group of PTSs, first proposed in [Berardi 1988], is specially suitable for this interpretation, and known as the 'Logic Cube'[7].

1.2.4. DEFINITION. **Logic Cube** The *cube of logical typed lambda calculi* consists of the following eight PTS$_\beta$s. Each of them has

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

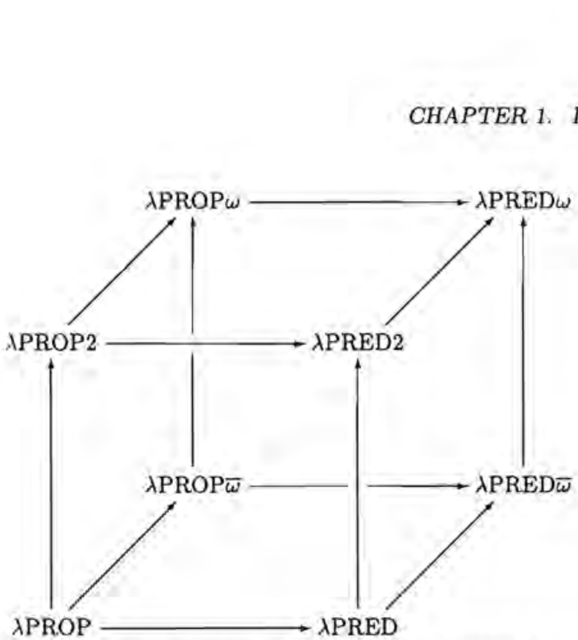
$$\mathcal{A}^{Type} = Prop : Type^p, Set : Type^s.$$

The rules of the systems are given by the following list. Since for these PTSs all rules $(s_1, s_2, s_3) \in \mathcal{R}$ have identical second and third arguments ($s_2 \equiv s_3$), we will henceforth abbreviate them as (s_1, s_2).

$\lambda PROP$	$(Prop, Prop)$
$\lambda PROP2$	$(Prop, Prop),\ (Type^p, Prop)$
$\lambda PROP\underline{\omega}$	$(Prop, Prop),\ (Type^p, Type^p)$
$\lambda PROP\omega$	$(Prop, Prop),\ (Type^p, Prop),\ (Type^p, Type^p)$
$\lambda PRED$	$(Prop, Prop),\ (Set, Set),\ (Set, Prop),\ (Set, Type^p)$
$\lambda PRED2$	$(Prop, Prop),\ (Set, Set),\ (Set, Prop),\ (Set, Type^p),\ (Type^p, Prop)$
$\lambda PRED\underline{\omega}$	$(Prop, Prop),\ (Set, Set),\ (Set, Prop),\ (Set, Type^p),\ (Type^p, Set),\ (Type^p, Type^p)$
$\lambda PRED\omega$	$(Prop, Prop),\ (Set, Set),\ (Set, Prop),\ (Set, Type^p),\ (Type^p, Set),\ (Type^p, Type^p),\ (Type^p, Prop)$

[6] Also known as the Curry-Howard-De Bruijn isomorphism , see [Nederpelt 1990].

[7] We follow the terminology of [Geuvers 1993]. In [Barendregt 1991] these systems are referred to as the 'L-cube', there the term 'Logic Cube' is reserved for the actual logics to which these PTSs correspond.



(In the diagram an arrow denotes inclusion of one system in another.)

The sorts of the PTSs in the Logic Cube are named in accordance with their role in the propositions-as-types interpretation: *Prop* is to denote the class of all propositions and *Set* the class of all sets, hence $A : Prop$ and $B : Set$ can be read as 'A is a proposition' and 'B is a set' respectively. As the type axioms ($Prop : Type^p$, $Set : Type^s$) show, $Type^p$ and $Type^s$ are supposed to serve as the 'types' of *Prop* and *Set*, and of the constructs built from *Prop* and *Set* by means of the rules.

The only PTS-rule that yields such constructs is *product*. Contrary to most standard presentations of logic, the derivation of a 'product' by means of a rule $(s_1, s_2, s_3) \in \mathcal{R}$ corresponds to showing *inside the formalism* that the particular product is a well-formed expression of the language. Only after a formula has been shown to be well-formed, we can create an inhabitant for it (using *abstraction* and *application*).

Viewed in this way the basic rule $(Prop, Prop)$ allows the construction of a 'product proposition', $\Pi x : A.B$, from two propositions A and B:

$$\frac{\Gamma \vdash A : Prop \quad \Gamma,\ x : A \vdash B : Prop}{\Gamma \vdash (\Pi x : A.B) : Prop}$$

Because $x \notin FV(B)$ in this case, $\Pi x : A.B$ can be seen as the implication $A \supset B$ in logic. This is confirmed by the introduction and elimination procedures for $\Pi x : A.B$ prescribed by *abstraction* and *application* respectively: in order to prove $\Pi x : A.B$ we must show that given an inhabitant of A ($x : A$ in the context), we can find an inhabitant of B. From a proof term for $\Pi x : A.B$, we obtain a proof (term) of B by applying it to a proof term for A.

In the same way (Set, Set) lets one define 'simple functional domains'. $(Set, Prop)$ corresponds to first order quantification over sets ($\forall x : A.\varphi$, with $A : Set$), and second order quantification ($\forall \alpha : Prop.\varphi$) is introduced by $(Type^p, Prop)$. Note that we write '$\forall$' instead of 'Π' in such cases, for obvious reasons. Furthermore, $(Set, Type^p)$ allows the formation of predicates as 'functions' from sets to propositions: e.g. terms P of type $\Pi x : A.Prop$ act as predicates in the sense that applying them to an element a of the set A yields a proposition Pa of type *Prop*). Finally $(Type^p, Type^p)$ makes it possible to construct 'functions' from *Prop* to *Prop* (and so on), which could for instance be used to include the logical connective for negation as a constant representing a function from propositions to propositions: $\neg : Prop \rightarrow Prop$.

If we call terms in *Prop* and *Set* ($A : Prop, B : Set$) *types* and terms in $Type^p$ and $Type^s$ ($C : Type^p, D : Type^s$) *kinds*, the arrangement of the PTS in the cube can be understood as arising from the dependencies between *types* and *kinds*; the possibilities for abstracting over terms of a specific category to form a term of another category.

In the standard orientation of the Logic Cube the system $\lambda PROP$, the smallest system, lies in the origin. There we can use the rule $(Prop, Prop)$ to abstract over a *type* to define a term that is again a *type*; 'types depending on types'. We can now think of the other three possible dependencies between *types* and *kinds* as spanning the cube, by viewing them as spatial dimensions:

Up: *types* depending on *kinds*, $(Type^p, Prop)$
Right: *kinds* depending on *types*, $(Set, Type^p)$
Backward: *kinds* depending on *kinds*, $(Type^p, Type^p)$

Each typed λ-calculus (λL) in the Logic Cube corresponds to a systems (L) of *intuitionistic* logic.

$PROP$ proposition logic
$PROP2$ second order proposition logic
$PROP\overline{\omega}$ weakly higher order proposition logic
$PROP\omega$ higher order proposition logic
$PRED$ predicate logic
$PRED2$ second order predicate logic
$PRED\overline{\omega}$ weakly higher order predicate logic
$PRED\omega$ higher order predicate logic

All these logics are minimal in the sense that the only logical operators are $\supset$ and $\forall$. Weakly higher order logics have variables for higher order propositions or predicates but no quantification over them. In the second and higher order systems it is possible to quantify over propositions. This feature enables us to define all the usual logical connectives in terms of '$\forall$' and '$\supset$':

$$\begin{aligned} A \wedge B &:= \forall \alpha : Prop.(A \supset (B \supset \alpha)) \supset \alpha \\ A \vee B &:= \forall \alpha : Prop.((A \supset \alpha) \supset (B \supset \alpha)) \supset \alpha \\ \bot &:= \forall \alpha : Prop.\alpha \\ \neg A &:= A \supset \bot \end{aligned}$$

The second order quantification also lets us express the rule of double negation elimination as an axiom schema: $\forall \alpha : Prop.((\alpha \supset \bot) \supset \bot) \supset \alpha$.

By adding this axiom to the context of the type derivation in PTSs, using some variable as inhabitant, every needed instance (e. g. $\neg\neg A \supset A$) can be derived by eliminating the quantification in the axiom schema with the appropriate type ($A : Prop$) using *application*.

In second (and higher) order systems with this axiom, *classical* propositional and predicate logics can be interpreted[8]. Classical versions of the first order systems (lower plane of the cube) are also possible but they require the introduction of '$\perp$' as a constant, the Ex Falso Sequitur Quodlibet rule, and the double negation rule in the type system (see [Geuvers 1993]).

1.2.2. Propositional logic in the propositions-as-types-interpretation

To give the reader some initial idea about the interpretation of logics in PTSs, we now give a simple example of a type derivation in $\lambda PROP$ proving the propositional tautology $(A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$. By comparing the type derivation with a natural deduction proof of this formula in $\Box PROP_{Fitch}$, we point out the type theoretical analogons of some of the main ingredients of Fitch-style deduction. This will give us a point of departure for the introduction of 'modal' PTSs.

In $\Box PROP_{Fitch}$ the tautology can be proved as follows:

1.	$A \supset (B \supset C)$	
2.	$\quad A \supset B$	
3.	$\quad\quad A$	
4.	$\quad\quad A \supset B$	(reiteration 2)
5.	$\quad\quad B$	($\supset$-elim 3,4)
6.	$\quad\quad A \supset (B \supset C)$	(reiteration 1)
7.	$\quad\quad B \supset C$	($\supset$-elim 3,6)
8.	$\quad\quad C$	($\supset$-elim 5,7)
9.	$\quad A \supset C$	($\supset$-intro 3-8)
10.	$(A \supset B) \supset (A \supset C)$	($\supset$-intro 2-9)
11.	$(A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$	($\supset$-intro 1-10)

This natural deduction proof corresponds to the term $\lambda x : A \supset (B \supset C).\lambda y : A \supset B.\lambda z : A.xz(yz)$ in $\lambda PROP$.

That this term is an inhabitant of the type $(A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$ can be derived as follows (assuming that $A \supset (B \supset C) : Prop$, $A \supset B : Prop$ and $A : Prop$ are already valid in Γ):

[8] See [Geuvers 1988].

0.	$\Gamma \vdash A \supset (B \supset C) : Prop$	(Start lem.)
1.	$\Gamma, x : A \supset (B \supset C) \vdash x : A \supset (B \supset C)$	(start 0)
1′	$\Gamma, x : A \supset (B \supset C) \vdash A \supset B : Prop$	(Start lem.)
2.	$\Gamma, x : A \supset (B \supset C), y : A \supset B \vdash y : A \supset B$	(start 1′)
2′	$\Gamma, x : A \supset (B \supset C), y : A \supset B \vdash A : Prop$	(Start lem.)
3.	$\Gamma, x : A \supset (B \supset C), y : A \supset B, z : A \vdash z : A$	(start 2′)
4.	$\Gamma, x : A \supset (B \supset C), y : A \supset B, z : A \vdash y : A \supset B$	(Start lem.)
5.	$\Gamma, x : A \supset (B \supset C), y : A \supset B, z : A \vdash yz : B$	(appl. 3,4)
6.	$\Gamma, x : A \supset (B \supset C), y : A \supset B, z : A \vdash x : A \supset (B \supset C)$	(Start lem.)
7.	$\Gamma, x : A \supset (B \supset C), y : A \supset B, z : A \vdash xz : (B \supset C)$	(appl. 3,6)
8.	$\Gamma, x : A \supset (B \supset C), y : A \supset B, z : A \vdash xz(yz) : C$	(appl. 5,7)
9.	$\Gamma, x : A \supset (B \supset C), y : A \supset B \vdash \lambda z : A.xz(yz) : A \supset C$	(abstr. 3,8)
10.	$\Gamma, x : A \supset (B \supset C) \vdash \lambda y : A \supset B.\lambda z : A.xz(yz) : (A \supset B) \supset (A \supset C)$	(abstr. 2,9)
11.	$\Gamma \vdash \lambda x : A \supset (B \supset C).\lambda y : A \supset B.\lambda z : A.xz(yz) :$ $(A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$	(abstr. 1,10)

In order to prove $(A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$, we must find an inhabitant (proof object) of this type. Given the analogy between *abstraction* and $\supset$-intro and *application* and $\supset$-elim, we can adopt the same strategy as in the natural deduction proof: first we simplify the problem to finding a term of type $(A \supset B) \supset (A \supset C)$ in a context extended with a 'fresh' object (x) of type $(A \supset (B \supset C))$ (line 1). By the same reasoning we can reduce the problem further to that of finding a proof object for C, in context which is extended with objects (y, and z) inhabiting $A \supset B$ and A respectively (line 2-3). Clearly we are done if we have proof terms for B and $B \supset C$. These can be found by combining the statements available in the context: applying the proof term x for $A \supset (B \supset C)$ to the proof term z for A yields a term xz proving $B \supset C$ (line 7). Likewise, combining the proof y of $(A \supset B)$ with z results in an inhabitant yz of B (line 5). Applying the new found inhabitants gives us the desired proof term, $xz(yz)$ for C (line 8). We end the derivation by discharging the statements added to the context through subsequent applications of *abstraction* (line 9-11).

To bring out the analogy between Fitch-style deduction and derivations in PTSs, we have numbered the lines in the type derivation in such a way that the type of a statement derivable in line i corresponds to the proposition occurring at line i in the natural deduction proof. Although some extra steps (0,1′ and 2′) are needed in the type derivation, it is easy to see how the addition of the statements $x : (A \supset (B \supset C))$, $y : (A \supset B)$, and $z : A$ to the context Γ corresponds to starting the subordinate proofs in the proof figure (lines 1, 2, and 3). Just as the hypotheses are discharged at the end of the subordinate proof by an application of $\supset$-I, the statements are removed from the context by an application of *abstraction* (lines 9,10,11) . Moreover, the nesting of the subordinate proofs in the proof figure is reflected in the order of the additional statements in the context of the type derivation: both proofs deal with the assumptions on a 'last in, first out' basis.

There is no rule corresponding directly to *reiteration* in natural deduction. This also becomes clear when comparing lines 4 and 6 in the natural deduction proof with lines 4 and 6 in the type derivation. There a meta theoretical property of PTSs, the Start lemma, is invoked.

1.2.5. LEMMA. **Start lemma**
Let Γ be a (legal) context, then $(x : A) \in \Gamma \Rightarrow \Gamma \vdash x : A$

This lemma tells us that a statement that is an element of a context is derivable on that context *regardless of its position* in that context (the proof uses induction and the rules *start* and *weakening*, given in the definition of PTSs). Hence a statement added to the context at some stage in the derivation will still be derivable when this context is extended with further statements. Since 'being derivable in a context' is the type theoretical analogon of 'occurring in a proof', the effect of the lemma is that 'hypotheses' can be 'reiterated' in type derivations.

1.2.3. Modal rules for K

In section 1.2 it was shown how Fitch-style deduction treats modal logic by extending propositional deduction with a new sort of proofs. If we want to use this idea to interpret modal logic in type theory, a type theoretical version of *strict* subordinate proofs is needed. In the framework of such a procedure the import rules and export rules of natural deduction (which depend on the general form of propositions) can be 'translated' into type theoretical rules depending on the general form of types.

In the preceding section we have seen that in the interpretation of propositional natural deduction the starting of a subordinate proof from an assumption corresponds to adding a statement to the context. Strict subordinate proofs differ from subordinate proofs in two respects: they require no hypotheses, and only formulas of a certain form may be reiterated in them. In formulating a type theoretical analogon to strict subordinate proofs the second requirement causes a problem; it implies that strict subordinate proofs correspond to derivations in a context in which (of the statements originally present) only statements with types of a certain form are available. In other words, strict subordinate proofs take place in a different context.

That this is not just a figure of speech follows from the above discussion of reiteration in typed λ-calculus. The Start lemma shows that in PTSs any statement that appears in the 'main proof' will automatically be available in any 'subordinate proof'. Therefore reiteration can not be restricted to statements of a certain form as long as the derivation is carried out in the (augmented) context of the main proof. A solution to this problem is to let the type theoretical counterparts of the import and export rules exchange statements between the original context of the derivation and a new related context. In this new context only certain statements that are derivable in the original context could then be declared derivable, depending on their type.

This solution requires a broadening of the notion of context: if Γ and Γ' are PTS-contexts then Γ ⧈ Γ' is a 'generalized context', in which Γ is to be called the 'main context' and Γ' the 'subordinate context'. The symbol '⧈' syntactically denotes that Γ and Γ' are in the subordination relation. Since subordinate contexts are to play the part of the strict subordinate proofs of modal natural deduction, we have to allow that every subordinate context can have its own subordinate context to an arbitrary depth. This means that things like 'Γ ⧈ Γ' ⧈ ε ⧈ Γ'' ⧈ ε' have to count as generalized contexts.

Given the extended set of contexts the K-import and -export rules can be expressed type theoretically. The K-import rule should state that from any statement with a type of the form $\Box B$ derivable in a generalized context G, a context G ⧈ ε can be generated where a statement of type B is derivable. This is the start of a strict subordinate proof; the subordinate context is empty. Furthermore any such statement with the type of the form $\Box B$, derivable the main context may be repeated in the subordinate context with its type in the form B (import in an existing strict subordinate proof). Hence the basic type theoretical import rule is:

$$K\ \mathit{import} : \frac{G \vdash A : \Box B}{G \boxdot \varepsilon \vdash \check{k}A : B.}$$

Consequently the K-export rule should say that a statement with type B derivable in the subordinate context may be reiterated in the main context with its type in the form $\Box B$:

$$K\ \mathit{export} : \frac{G \boxdot \varepsilon \vdash A : B}{G \vdash \hat{k}A : \Box B.}$$

Note that in this case the subordinate context has to be empty; the proof of B must be categorical (all assumptions discharged).

So far we have only discussed modal deduction in terms of types and contexts but, as the above rules show, there is another aspect to be considered: the proof objects. Since the types of the statements are changed upon K-import and K-export it is clear that the original proof objects cannot be left unchanged: a proof object of type $\Box B$ in the main context cannot simply be assumed to be an inhabitant of type B in the subordinate context. Yet the formulation of the K-import and -export rules suggests that there is a simple relation between the proof object before and after the application of the rules, the proof object (A) is transformed into a proof object consisting of the original object with a function ($\check{k}$ or $\hat{k}$) applied to it. The function $\check{k}$ 'specializes' a proof of B in all accessible worlds (i.e. a proof of $\Box B$) to a proof of B in one (arbitrary) world, the function $\hat{k}$ 'generalizes' a categorical proof of B in an arbitrary accessible world to a proof of B in all accessible worlds.

The use of these functions garantees that the 'modal' steps in the proof of a proposition are represented in its proof object. The natural deduction proof of the K-axiom ($\Box(A \supset B) \supset (\Box A \supset \Box B)$), given in section 1.1.2, corresponds to the term $\lambda x : \Box(A \supset B).(\lambda y : \Box A.\hat{k}(\check{k}x(\check{k}y)))$. The '$\check{k}$s' and '$\hat{k}$s' appearing in this term signify applications of K-import and K-export. To illustrate the use of the modal type theoretical rules, we prove that the above term is of type $\Box(A \supset B) \supset (\Box A \supset \Box B)$ (assuming that $\Gamma \vdash \Box(A \supset B) : \mathit{Prop}$ and $\Gamma \vdash \Box A : \mathit{Prop}$):

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

Modal Pure Type Systems

We now give a formal definition of the 'modal' type systems. They will be called *Modal Pure Type Systems* (MPTSs), and they differ from PTSs in three respects:

Additional (modal) terms

A generalized notion of context

Additional (modal) rules

We will start by giving definitions of these extensions. First the set of pseudoterms has to be extended with 'modal types' ($\Box\mathcal{T}$) and 'modal proof terms' ($\check{k}\mathcal{T}$, $\hat{k}\mathcal{T}$).

1.2.6. Definition. **Pseudoterms**
For $\mathcal{S}$ some set, the set of pseudoterms over $\mathcal{S}$, $\mathcal{T}$ is

$$\mathcal{T} ::= \mathcal{S}|\,Var|\Pi\,Var : \mathcal{T}.\mathcal{T})|(\lambda\,Var : \mathcal{T}.\mathcal{T})|\mathcal{T}\mathcal{T}|\Box\mathcal{T}|\check{k}\mathcal{T}|\hat{k}\mathcal{T}|\mathcal{C},$$

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

The extended notion of context in MPTS is defined starting from the PTS-definition of pseudocontext.

1.2.7. Definition. **Generalized 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: $x_1 : A_1, \ldots, x_n : A_n$. The empty context is denoted by ε. If $\Gamma = x_1 : A_1, \ldots, x_n : A_n$ then $\Gamma, x : B = x_1 : A_1, \ldots, x_n : A_n, x : B$

(iiia) A *generalized pseudo-context* is a finite ordered sequence of pseudo-contexts and separators $\boxdot$, in which all variables occuring as subject are different: $G = \Gamma_1 \boxdot \ldots \boxdot \Gamma_n$
If $G = x_1 : A_1, \ldots \boxdot \ldots \boxdot\ x_m : A_m, \ldots, x_n : A_n$
then $G, x : B = x_1 : A_1, \ldots \boxdot \ldots \boxdot x_m : A_m, \ldots, x_n : A_n, x : B$,
and $G \boxdot \varepsilon = x_1 : A_1, \ldots \boxdot \ldots \boxdot x_m : A_m, \ldots, x_n : A_n \boxdot \varepsilon$

(iiib) Alternatively an *inductive definition* of the set of generalized pseudo contexts $\mathcal{G}$ can be given based on the set $\mathcal{O}$ of PTS-contexts: $\mathcal{G} = \mathcal{O} \mid \mathcal{G} \boxdot \mathcal{O}$

We take $\Gamma, \Gamma', \ldots$ to be ranging over $\mathcal{O}$ and $G, G', \ldots$ to range over $\mathcal{G}$.

Given the following notational device, we can complete the definition of MPTSs.

Notation. $G \vdash A : B : C$ means $G \vdash A : B$ and $G \vdash B : C$.

1.2.8. DEFINITION. **Modal Pure Type Systems**

A *Modal Pure Type System with β-conversion*, MPTS_β, is given by a set $\mathcal{S}$ of *sorts* containing *Prop*, *Set*, and *Type*, a set $\mathcal{A}^{Type} \subset \mathcal{S} \times \mathcal{S}$ of *typing axioms*, a set $\mathcal{A}^{Logic} \subset \mathcal{C} \times \mathcal{T}$ 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 $\Box\lambda_\beta(\mathcal{S}, \mathcal{A}, \mathcal{R})$ and is the typed λ-calculus with the following deduction rules:

$$(axiom)\quad \varepsilon \vdash s_1 : s_2 \quad \text{if } s_1 : s_2 \in \mathcal{A}^{type} \qquad \varepsilon \vdash c : A : Prop \quad \text{if } c : A \in \mathcal{A}^{Logic}$$

$$(start)\quad \frac{G \vdash A : s}{G,\ x : A \vdash x : A}$$

$$(weakening)\quad \frac{G \vdash A : B \quad G \vdash C : s}{G,\ x : C \vdash A : B}$$

$$(product)\quad \frac{G \vdash A : s_1 \quad G,\ x : A \vdash B : s_2}{G \vdash (\Pi x : A.B) : s_3} \quad \text{if } (s_1, s_2, s_3) \in \mathcal{R}$$

$$(application)\quad \frac{G \vdash F : (\Pi x : A.B) \quad G \vdash a : A}{G \vdash Fa : B[x := a]}$$

$$(abstraction)\quad \frac{G,\ x : A \vdash b : B \quad G \vdash (\Pi x : A.B) : s}{G \vdash (\lambda x : A.b) : (\Pi x : A.B)}$$

$$(conversion)\quad \frac{G \vdash A : B \quad G \vdash B' : s \quad B =_\beta B'}{G \vdash A : B'}$$

$$(boxing)\quad \frac{G \vdash A : Prop}{G \vdash \Box A : Prop}$$

$$(transfer_1)\quad \frac{G \vdash A : s}{G \boxdot \varepsilon \vdash A : s}$$

$$(transfer_2)\quad \frac{G \vdash A : B : Type}{G \boxdot \varepsilon \vdash A : B}$$

$$(transfer_3)\quad \frac{G \vdash A : B : Set}{G \boxdot \varepsilon \vdash A : B}$$

$$(transfer_{ax})\quad \frac{G \vdash c : A : Prop}{G \boxdot \varepsilon \vdash c : A}$$

$$(K\ import)\quad \frac{G \vdash A : \Box B : Prop}{G \boxdot \varepsilon \vdash \check{k}A : B}$$

$$(K\ export)\quad \frac{G \boxdot \varepsilon \vdash A : B : Prop}{G \vdash \hat{k}A : \Box B}$$

where s ranges over the set of sorts $\mathcal{S}$, c over the set of constants $\mathcal{C}$, x ranges over variables, and it is assumed that in the rules (*start*) and (*weakening*) the newly declared variable x is always fresh, i.e. that is it has not yet been declared in G.

The rules up to *conversion* are familiar, they are the PTS-rules stated for generalized contexts (G). This means that the original rules for type derivation hold in all subordinate contexts regardless of their 'modal depth', just like the propositional deduction rules in $\Box PROP_{fitch}$ hold in all (strict) subordinate proofs.

The K-import and K-export rule are defined somewhat differently than in the previous section. For both rules we additionally demand that the type of the premiss ($\Box B$ for import, B for export) is itself of type *Prop*. In this way the K-export rule together with the *boxing* rule ensures that no terms are 'modalized' other than terms of type *Prop*. Formulating the K-rules like this also facilitates the proofs of meta-theoretical properties of MPTSs.

The import and export rules deal with the transport of statements $A : B : Prop$ between main and subordinate context, where the term A represents a proof of the proposition B. For these statements, reiteration is restricted in a modal system. However, besides these proof/proposition statements there are many other kinds of statements in MPTSs. The *transfer*-rules make sure that all of these can be reiterated freely in all subordinate contexts. In this way any derivable statement that a certain term (A) is a set or a proposition ($A : Set, A : Prop$), will be derivable in a subordinate context by means of $transfer_1$. Likewise $transfer_2$ makes predicates ($A : B \rightarrow Prop : Type^p$) available in subordinate contexts. $Transfer_3$ does the same for elements of sets, and $transfer_{ax}$-rule for 'logical axioms'. The combined effect of the *transfer*-rules is that for all terms not representing proofs in MPTSs the extra structure of the generalized contexts is irrelevant, they behave just like they do in PTSs[9]. This corresponds to the meta theoretical assumption in $\Box PROP_{fitch}$ that the language (the set of proposition letters) is the same for all (strict) subordinate proofs.

In PTSs we only had axioms of the form $s_1 : s_2$, where $s_1, s_2 \in S$. We shall call these axioms 'Typing axioms' (elements of $\mathcal{A}^{Type}$), to distinguish them from the 'Logical axioms' (elements of $\mathcal{A}^{Logic}$) like the law of double negation and additional modal axioms that can be expressed in the type theoretical language.

There is no need to formally define the use of logical axioms in PTSs, in that framework one postulates a variable inhabiting the axiom, e. g. $x : \forall\alpha : Prop.((\alpha \supset \bot) \supset \bot) \supset \alpha$, and adds this statement to the context. The axiom can then be used throughout the type derivation, after which it remains in the context as would an undischarged assumption. In MPTSs, however, it has to be ensured that the logical axioms are also derivable in the subordinate context. Hence postulating an inhabiting variable for an axiom is not enough; we also need a way to propagate these variables to subordinate contexts. This creates a problem: such an inhabiting variable is a term representing a proof of a proposition ($A : B : Prop$). Therefore any form of 'transfer' for these axiomatic statements bring us into conflict with the basic idea that only terms representing proofs of propositions of a certain modal form ($\Box B$) may be imported into a subordinate context.

To maintain a manageable distinction between terms for which the import and export rules apply (terms representing proofs) and terms for which the transfer rules apply, we have opted for an explicit treatment of the logical axioms. Each MPTS has a (possibly empty) set of logical axioms $\mathcal{A}^{Logic}$. The axioms all have inhabiting constants c, elements of a set of constants C. For instance all 'classical' MPTSs have the double negation axiom $c : \forall\alpha.((\alpha \supset \bot) \supset \bot) \supset \alpha$ as an element of $\mathcal{A}^{Logic}$. This use of constants allows us to treat the logical axioms exactly like the typing axioms. Therefore the *axiom*-rule of the PTSs has

[9] See Block insertion and deletion lemma, chapter 3.4.

been extended by a clause declaring that the logic axioms are derivable in the empty context:

$$\varepsilon \vdash c : A : Prop \quad \text{if } c : A \in \mathcal{A}^{Logic}$$

A transfer rule ensures that the logical axioms will be derivable in any subordinate context (like $transfer_1$ does for the typing axioms):

$$(transfer_{ax}) \quad \frac{G \vdash c : A : Prop}{G \boxdot \varepsilon \vdash c : A}$$

In analogy with the Logic Cube we now define a cube of modal type systems

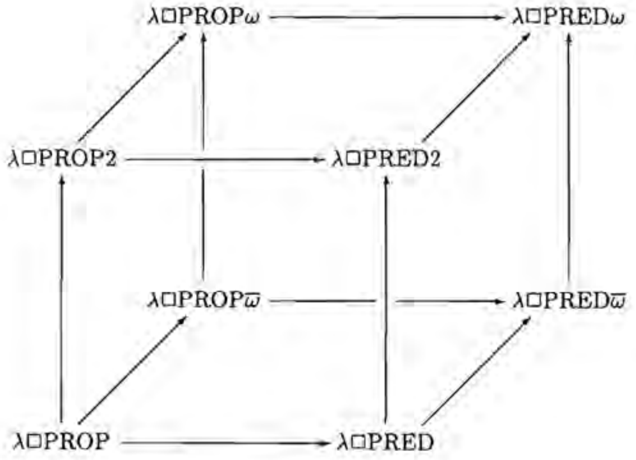
1.2.9. DEFINITION. **Modal Logic Cube**
The *cube of modal logical typed lambda calculi* consists of the following eight MPTS_βs. Each of them has

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

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

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

The rules $\mathcal{R}$ of an MPTS $\lambda\Box S$ in the Modal Logic Cube are the same as those of the PTS λS in the Logic Cube, e.g. like $\lambda PROP$, $\lambda\Box PROP$ has $\mathcal{R} = (Prop, Prop)$.



1.2.4. Extending the MPTSs

Just like in modal natural deduction there are two options regarding the strengthening of the modal type systems: adding axioms and adding rules.

Extension by axioms

Modal axioms can be added to the second order MPTSs (in the upper plane of the Modal Logic Cube) rather straightforwardly because the quantification over types (propositions) enables the formulation of axiom schemes in the type theoretical language:

$c_d : (\forall\alpha : Prop.(\Box\alpha \supset \neg\Box\neg\alpha))$

$c_t : (\forall\alpha : Prop.(\Box\alpha \supset \alpha))$

$c_b : (\forall\alpha : Prop.(\alpha \supset \Box\neg\Box\neg\alpha))$

$c_4 : (\forall\alpha : Prop.(\Box\alpha \supset \Box\Box\alpha))$

$c_5 : (\forall\alpha : Prop.(\neg\Box\alpha \supset \Box\neg\Box\alpha))$

To obtain a deduction system for normal modal logics extending K, one or more axiom schemes (along with an inhabiting constant) are added to the set of logical axioms $\mathcal{A}^{Logic}$. The rules *axioms*, *weakening* and $transfer_{ax}$ will then make sure that these schemas are derivable on any generalized context[10].

During the deduction the needed axioms can be derived from these schemas by means of the $(\forall E)$-rule, for instance to prove A from the assumption $\Box A$ in a context Γ using the axiom schema T:

$$\begin{array}{lll}
1. & \Gamma, x : \Box A \vdash x : \Box A & \\
2. & \Gamma, x : \Box A \vdash c_t : \forall\alpha : Prop.(\Box\alpha \supset \alpha) & (axiom) \\
3. & \Gamma, x : \Box A \vdash A : Prop & (A : Prop \in \Gamma) \\
4. & \Gamma, x : \Box A \vdash c_t A : (\Box A \supset A) & \\
5. & \Gamma, x : \Box A \vdash (c_t A)x : A &
\end{array}$$

Hence, this extension works just like the Fitch-style extension by axioms with one difference: instead of writing an axiom at any point in a proof, an axiom schema may be written at any point in a proof.

Extension by rules

The extension of a MPTS by means of rules to accommodate normal systems stronger than K is completely analogous to the extension by rules of the Fitch-style deduction: extra rules are given for the import and export from statements into, respectively out of, the subordinate context. Again a single *import rule* for each of the axioms 4, 5, and B, a single *export rule* for the axioms D and T. Each of these rules introduces its own function, a 'check'-function for the import rules and a 'hat'-function for the export rules (we add $\check{4}\mathcal{T}|\check{5}\mathcal{T}|\check{b}\mathcal{T}|\hat{d}\mathcal{T}|\hat{t}\mathcal{T}$ to the set $\mathcal{T}$ of pseudo terms), introducing a new connection between proofs in the main context and the subordinate context.

$$4\,import : \frac{G \vdash A : \Box B : Prop}{G \boxdot \varepsilon \vdash \check{4}A : \Box B}$$

[10]See proof of Start-lemma chapter 3

$$5\,import : \frac{G \vdash A : \neg\Box B : Prop}{G \boxdot \varepsilon \vdash \check{5}A : \neg\Box B}$$

$$B\,import : \frac{G \vdash A : B : Prop}{G \boxdot \varepsilon \vdash \check{b}A : \neg\Box\neg B}$$

$$D\,export : \frac{G \boxdot \varepsilon \vdash A : B : Prop}{G \vdash \hat{d}A : \neg\Box\neg B}$$

$$T\,export : \frac{G \boxdot \varepsilon \vdash A : B : Prop}{G \vdash \hat{t}A : B}$$

From axioms to rules

For each of the modal axioms inhabited by constants in the extension by axioms, we can find an inhabitant in the extension by rules. For the 4, 5 and B axiom, this inhabitant is derived using the corresponding import rule in combination with K-export. We show this for the 4 axiom:

1. $\alpha : Prop, x : \Box\alpha \vdash x : \Box\alpha$
2. $\alpha : Prop, x : \Box\alpha \boxdot \varepsilon \vdash \check{4}x : \Box\alpha$ (4-import 1)
3. $\alpha : Prop, x : \Box\alpha \vdash \hat{k}(\check{4}x) : \Box\Box\alpha$ (K-export 2)
4. $\alpha : Prop \vdash \lambda x : \Box\alpha.\hat{k}(\check{4}x) : \Box\alpha \supset \Box\Box\alpha$
5. $\varepsilon \vdash \lambda\alpha : Prop.(\lambda x : \Box\alpha.\hat{k}(\check{4}x)) : \forall\alpha : Prop.(\Box\alpha \supset \Box\Box\alpha))$

The D and T axioms are proved using the corresponding export rule in combination with K-import. For instance the T-axiom:

1. $\alpha : Prop, x : \Box\alpha \vdash x : \Box\alpha$
2. $\alpha : Prop, x : \Box\alpha \boxdot \varepsilon \vdash \check{k}x : \alpha$ (K-import 1)
3. $\alpha : Prop, x : \Box\alpha \vdash \hat{t}(\check{k}x) : \alpha$ (T-export 2)
4. $\alpha : Prop \vdash \lambda x : \Box\alpha.\hat{t}(\check{k}x) : \Box\alpha \supset \alpha$
5. $\varepsilon \vdash \lambda\alpha : Prop.(\lambda x : \Box\alpha.\hat{t}(\check{k}x)) : \forall\alpha : Prop.(\Box\alpha \supset \alpha))$

If we distinguish between the two extensions by indexing the derivation sign ('$\vdash^{ax}$' for extension by axioms, '$\vdash^{ru}$' for extension by rules), the situation can be summarized as follows:

$G \vdash^{ax} M : \varphi \Rightarrow G \vdash^{ru} \overline{M} : \varphi$, where $\overline{M}$ is obtained from M by substituting

$\lambda\alpha : Prop.(\lambda x : \Box\alpha.\hat{k}(\check{4}x))$ for all occurrences of c_4,

$\lambda\alpha : Prop.(\lambda x : \neg\Box\alpha.\hat{k}(\check{5}x))$ for all occurrences of c_5,

$\lambda\alpha : Prop.(\lambda x : \alpha.\hat{k}(\check{b}x))$ for all occurrences of c_b,

$\lambda\alpha : Prop.(\lambda x : \Box\alpha.\hat{d}(\check{k}x))$ for all occurrences of c_d,

$\lambda\alpha : Prop.(\lambda x : \Box\alpha.\hat{t}(\check{k}x))$ for all occurrences of c_t.

From rules to axioms

To get from rules to axioms, we must know how the additional import and export rules can be 'mimicked' using the axiom schemas in $\mathcal{A}^{Logic}$ and the K-rules.

For the import rules, this can be done using the corresponding axiom schema and K-import. For instance, 5-import:

1.	$G \vdash M : \neg\Box A : Prop$	
2.	$G \vdash c_5 : (\forall\alpha : Prop.(\neg\Box\alpha \supset \Box\neg\Box\alpha))$	(5-axiom)
3.	$G \vdash A : Prop$	
4.	$G \vdash c_5 A : (\Box A \supset \Box\neg\Box A)$	
5.	$G \vdash c_5 AM : \Box\neg\Box A$	
6.	$G \boxdot \varepsilon \vdash \check{k}(c_5 AM) : \neg\Box A$	(K-import 5)

To achieve the effect of 5-import on $M : \neg\Box A$, we first derive an A-instance of the 5-axiomschema (lines 2-4). By applying it with $M : \neg\Box A$, we obtain a term of type $\Box\neg\Box A$. This statement is of the right form to apply K-import, and that yields the desired result: a term, based on M, of type $\neg\Box A$ on context $G \boxdot \varepsilon$. Hence $\check{k}(c_5 AM)$ is the 'axiomatic version' of $\check{5}M$. Note that the axiomatic term corresponding to $\check{5}M$ depends on the type of M, since the axiom schema has to be instantiated with the correct type for the above derivation to work.

To mimic the export rules, we need K-export and the corresponding axiom schema. For example D-export:

1.	$G \boxdot \varepsilon \vdash M : A : Prop$	
2.	$G \vdash \hat{k}M : \Box A$	(K-export 1)
3.	$G \vdash c_d : (\forall\alpha : Prop.(\Box\alpha \supset \neg\Box\neg\alpha))$	(D-axiom)
4.	$G \vdash A : Prop$	
5.	$G \vdash c_d A : (\Box A \supset \neg\Box\neg A)$	
6.	$G \vdash (c_d A)(\hat{k}M) : \neg\Box\neg A$	

Here we first apply K-export, resulting in a proof of $\Box A$ in context G, and then perform '$\supset$-elim' with an A-instance of the D-axiom. Hence the axiomatic version of $\hat{d}M$ is $(c_d A)(\hat{k}M)$ for M of type A.

Therefore, we have the following relation between terms in the extension by rules and terms in the extension by axioms: $G \vdash^{ru} M : \varphi \Rightarrow G \vdash^{ax} \underline{M} : \varphi$, where $\underline{M}$ is obtained from M by substituting

$\check{k}(c_4 AM)$ for all occurrences of $\hat{4}M$, where $M : \Box A$,

$\check{k}(c_5 AM)$ for all occurrences of $\check{5}M$, where $M : \neg\Box A$,

$\check{k}(c_b AM)$ for all occurrences of $\check{b}M$, where $M : A$,

$c_d AM(\hat{k}M)$ for all occurrences of $\hat{d}M$, where $M : A$,

$c_t AM(\hat{k}M)$ for all occurrences of $\hat{t}M$, where $M : A$.

Together with the substitution given earlier, this implies that the extension by axioms and the extension by rules are equivalent in the sense that for every inhabited type in the extension by rules we can find an inhabitant in the extension by axioms and vice versa. Note that this 'equi-inhabitability' does not yet give us a bijection between inhabitants in the two extensions[11].

In this thesis we concentrate on two MPTSs from the Modal Logic Cube: $\lambda\Box PROP2$ for the interpretation of modal propositional logics and $\lambda\Box PRED2$ for modal predicate logics. As defined, both systems correspond to the basic normal modal (propositional/predicate) logic K. For normal modal logics $KS_1 \ldots S_n$, we simply add the corresponding axioms or import/export rules.

[11]It can be shown that $\underline{(\overline{M})}$ is equal to M and that $\overline{(\underline{M})}$ is equal to M, but only modulo $\twoheadrightarrow_\beta$ and some modal reductions on terms that will be discussed in chapters 2 and 3.

Chapter 2

Interpreting modal logic

Following [Geuvers 1993], we map modal logic to typed λ-calculus by first defining a typed system (ΛL) as close as possible to the original logic (L) and then showing that this system is equivalent to the system (λL) in the Modal Logic Cube.

However, we cannot use the original Fitch-style formulation of modal propositional logic $\Box PROP_{fitch}$ since it has classical propositional logic underlying it. In the standard interpretation of non-modal first order (intuitionistic as well as classical) propositional logic, the Cube system $\lambda PROP2$ is used. It corresponds to second order intuitionistic propositional logic. The quantification over propositions makes it possible to define the usual propositional connectives $\wedge, \vee$, and $\leftrightarrow$, in terms of '$\forall$' and '$\supset$', and to express the double negation law as an axiom.

Therefore we will start the mapping by reformulating our modal logic as a second order intuitionistic logic $\Box PROP2$ and then follow the path:

$$\Box PROP_{fitch} \Rightarrow \Box PROP2 \Leftrightarrow \Lambda\Box PROP2 \Leftrightarrow \lambda\Box PROP2$$

to the MPTS $\lambda\Box PROP2$ through the intermediate system $\Lambda\Box PROP2$ (where the $\Rightarrow$ is the conservativity of $\Box PROP2$ over $\Box PROP_{fitch}$).

The structure of the chapter follows that of the mapping, the systems are introduced following the arrows from left to right. The chapter closes with a discussion of the possibilities for proof reductions in the modal typed lambda calculus system.

2.1. $\Box PROP2$

As in the Logic Cube, we move to second order intuitionistic logic to interpret modal (classical) propositional logic. This logic will be defined in a typed lambda calculus manner. The difference with presentation in [Geuvers 1993] is that we will be using Fitch-style instead of Gentzen-style deduction rules.

2.1.1. DEFINITION. **Language**

1 The *domain* is

$$\mathcal{D} ::= Prop,$$

2 The *terms* of the second order language are the following elements of *Prop*:

- There are countably many variables of domain *Prop*
- If $\varphi \in Prop$, x a variable of domain *Prop* then $(\forall x \in Prop.\varphi) \in Prop$,
- If $\varphi \in Prop$ and $\psi \in Prop$, then $\varphi \supset \psi \in Prop$
- If $\varphi \in Prop$, then $\Box\varphi \in Prop$

We adopt the terminology of [Geuvers 1993] and understand *terms* to denote the set of all expressions in the language of the system. The set of *formulas* is a subset of *terms*: all terms that are an element of *Prop*. Since *Prop* is the only domain in $\Box PROP2$, formulas and terms coincide for this logic. However, we will need the distinction later on, when we move to modal predicate logic.

In addition to terms and formulas, we need a third category of expressions: a *notification* is an expression of the form $t \in D$, where t is a term and D a domain of the logic. Since $\Box PROP2$ has *Prop* as its only domain, notifications in this logic are of the form $\varphi \in Prop$, where φ is a formula. Notifications are needed to deal explicitly with the variables and terms involved in $\forall$-intro and $\forall$-elim rules (to be defined below) in the natural deduction proofs.

The other logical connectives featuring in $\Box PROP_{fitch}$ are definable in $\Box PROP2$ using '$\forall$' and '$\supset$' (where φ and ψ are formulas):

$$\begin{aligned}
\varphi \wedge \psi &:= \forall x \in Prop.(\varphi \supset (\psi \supset x)) \supset x, \\
\varphi \vee \psi &:= \forall x \in Prop.(\varphi \supset x) \supset ((\psi \supset x) \supset x), \\
\bot &:= \forall x \in Prop.x, \\
\neg\varphi &:= \varphi \supset \bot,
\end{aligned}$$

In the same way the existential quantifier and the diamond operator can be defined:
$\exists x \in Prop.\varphi := \forall y \in Prop.(\forall x \in Prop.\varphi \supset y) \supset y$, $\Diamond\varphi := (\Box\varphi \supset \bot) \supset \bot$.

The underlying propositional logic of $\Box PROP2$ is *second order intuitionistic* logic. This requires two modifications of the deduction rules given for $\Box PROP_{fitch}$ (classical first order logic): we need quantification rules for the propositional variables, and a way to deal with the 'classical' elimination of double negation (the $\neg$-elim rule in $\Box PROP_{fitch}$).

Quantification will be treated using 'explicit declaration' of propositional variables. This means that in the $\forall$-intro rule the propositional variable (over which the quantification is to take place) is first introduced as an assumption with its own hypothesis interval. Likewise in the $\forall$-elim rule, the term with which the $\forall$-formula is instantiated is derived beforehand (by means of the new *term*-rule). This approach brings out the similarity between the $\forall$- and $\supset$-rules, taking us closer to typed λ-calculus. It is comparable with the 'x-general proofs' of Fitch ([Fitch 1952]) and with Nederpelts 'flag deduction' ([Nederpelt 1977]), and avoids the 'free occurrence administration' of the 'implicit' rules used in [Van Westrhenen et al. 1993].

Using the second order quantification, we can express the double negation rule ($\neg$-elim) as an axiom:

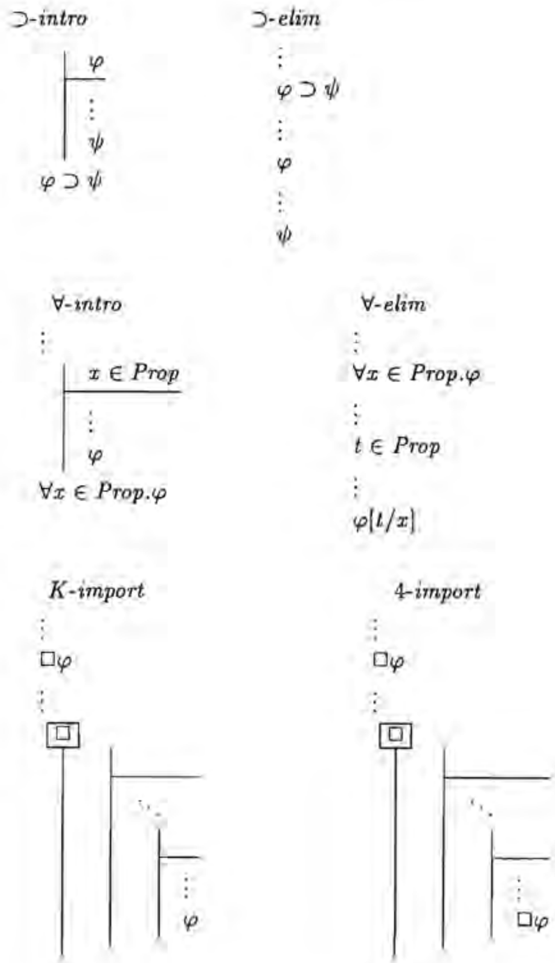
$$\forall x \in Prop.((x \supset \bot) \supset \bot) \supset x$$

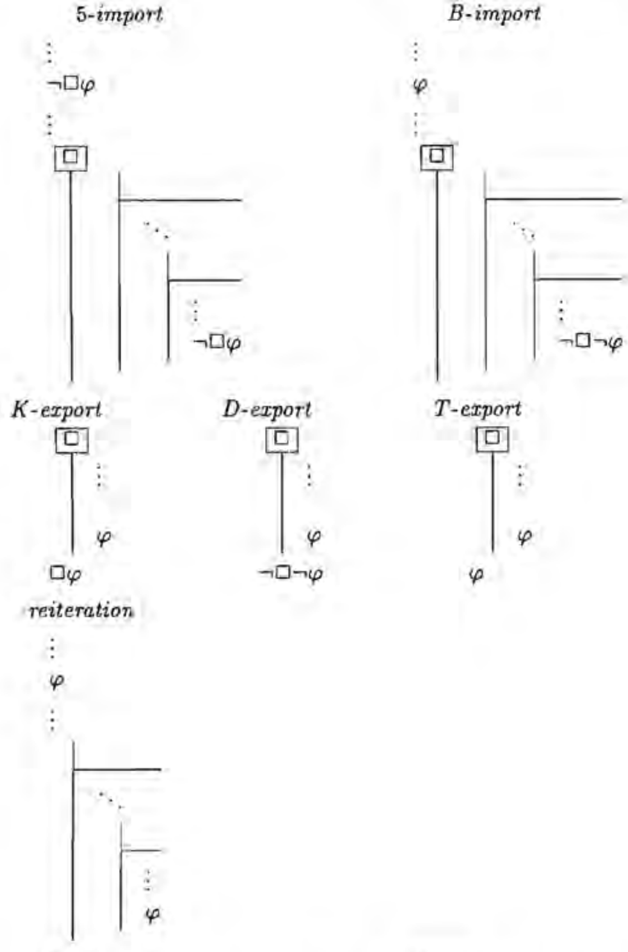
A rule is added to ensure that this axiom is instantly derivable anywhere in a proof.

The definition of *proof figure* of □*PROP2* is the same as for □$PROP_{fitch}$, with the difference that we now write expressions from the second order language in the proof figures.

2.1.2. DEFINITION. **Deduction rules**

For φ and ψ formulas of the second order language the Fitch-style *deduction rules* of □*PROP2* are:





term
$t \in Prop$

axiom
$\forall x \in Prop.((x \supset \bot) \supset \bot) \supset x$

2.1.3. Definition. **Application of deduction rules**
Given a proof figure **D**, with interval $D = [1, n]$, formulas and notifications $F_1, \ldots, F_n$ and hypotheses intervals **H**. A formula or notification 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 $R \in \{\forall\text{-}elim, \supset\text{-}elim)\}$.
In this case the premisses and the conclusion E all lie in the *same interval*. The order in the which the premisses appear may differ from the one given in the table.

2 $R = \supset\text{-}intro$.
There has to be a hypothesis-interval $[k, l] \in \mathbf{H}$, such that $F_k = \varphi$ and $F_l = \psi$. The conclusion $E = \varphi \supset \psi$ and the interval $[k, l]$ have to lie in the same interval.

3 $R = \forall$-*intro*.
There has to be a hypothesis-interval $[k, l] \in \mathbf{H}$, such that $F_k = x \in Prop$ and $F_l = \varphi$, where x is fresh variable (x does not occur in $F_1, \ldots, F_{k-1}$). The conclusion $E = \forall x \in Prop.\varphi$ and the interval $[k, l]$ have to lie in the same interval.

4 $R =$ *reiteration*.
If the premiss A lies in the interval $I \in \mathbf{I} \cup \{D\}$ and the conclusion $E = A$ lies in the interval $J \in \mathbf{I} \cup \{D\}$, 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 $gr(E)$.

5 $R =$ *term*.
If t is an element of *terms* the conclusion $E = t \in Prop$ may lie in any interval $I \in \mathbf{I}$.

6 $R =$ *axiom*.
The conclusion $E = \forall x \in Prop.(x \supset \bot) \supset \bot) \supset x$ may lie in any interval $I \in \mathbf{I}$.

7 $R = K$ *import*.
If the premiss $\Box\varphi$ lies in interval $I \in \mathbf{I}$ and the conclusion $E = \varphi$ lies in the interval $J \in \mathbf{I}$, then it has to be the case that *the interval* $J \subset I$ and there exists *exactly one* $K \in \mathbf{M}$ such that $J \subseteq K \subset I$. Or in terms of gr: the first coordinate of $gr(E)$ is the first coordinate of $gr(\Box\varphi) + 1$.

8 $R = K$ *export*.
If the premiss φ lies in interval $I \in \mathbf{M}$ and the conclusion $E = \Box\varphi$ lies in the interval $J \in \mathbf{I}$, then it has to be the case that *the interval* I lies *in the interval* J.

9 $R = 4$ *import*.
If the premiss $\Box\varphi$ lies in interval $I \in \mathbf{I}$ and the conclusion $E = \Box\varphi$ lies in the interval $J \in \mathbf{I}$, then it has to be the case that *the interval* $J \subset I$ and there exists *exactly one* $K \in \mathbf{M}$ such that $J \subseteq K \subset I$. Or in terms of gr: the first coordinate of $gr(E)$ is the first coordinate of $gr(\Box\varphi) + 1$.

10 $R = 5$ *import*.
If the premiss $\neg\Box\varphi$ lies in interval $I \in \mathbf{I}$ and the conclusion $E = \neg\Box\varphi$ lies in the interval $J \in \mathbf{I}$, then it has to be the case that *the interval* $J \subset I$ and there exists *exactly one* $K \in \mathbf{M}$ such that $J \subseteq K \subset I$. Or in terms of gr: the first coordinate of $gr(E)$ is the first coordinate of $gr(\Box\varphi) + 1$.

11 $R = B$ *import*.
If the premiss φ lies in interval $I \in \mathbf{I}$ and the conclusion $E = \neg\Box\neg\varphi$ lies in the interval $J \in \mathbf{I}$, then it has to be the case that *the interval* $J \subset I$ and there exists *exactly one* $K \in \mathbf{M}$ such that $J \subseteq K \subset I$. Or in terms of gr: the first coordinate of $gr(E)$ is the first coordinate of $gr(\Box\varphi) + 1$.

12 $R = D$ *export*.
If the premiss φ lies in interval $I \in \mathbf{M}$ and the conclusion $E = \neg\Box\neg\varphi$ lies in the interval $J \in \mathbf{I}$, then it has to be the case that *the interval* I lies *in the interval* J.

13 $R = T$ *export.*
If the premiss φ lies in interval $I \in \mathbf{M}$ and the conclusion $E = \varphi$ lies in the interval $J \in \mathbf{I}$, then it has to be the case that *the interval* I lies *in the interval* J.

Clauses 5 and 6 are meant to ensure that terms and the double negation axiom may be written at any line in a proof figure without further justification (cf. [Fitch 1952]).

Note that in this system each import-rule itself allows import 'over' any number of assumptions (hypothesis intervals) lying inside the modal subordinate proof. The import-rules of $\Box Prop_{fitch}$ are a special case under this formulation.

2.1.4. 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 or notification $F_i (1 \leq i \leq n)$ is a hypothesis or the result of the application of a deduction rule on a number of formulas or notifications preceding F_i.

2.1.5. 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 or notification $F_i (1 \leq i \leq n)$ is a hypothesis or the result of the application of a deduction rule on a number of formulas or notifications preceding F_i.

Derivability is a property of formulas, not of notifications. Hence we have to restrict the hypotheses in the definition of 'derivation with hypotheses' to *formulas*: derivations in which assumptions like '$x \in Prop$' are left open should not be allowed.

2.1.6. Definition. **Derivability**

1 A formula C is *derivable* (notation $\vdash C$) if there exists a derivation of C.

2 A formula C is *derivable* from the formulas $P_1, \dots P_m$ if there exists a derivation of C from $P_1 \dots P_m$; notation as $P_1, \dots, P_m \vdash C$.

3 Let $\Gamma \subseteq$ *terms* 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: $\Gamma \vdash C$. (Γ may be empty; notation $\phi \vdash C$ iff $\vdash C$).

Defined connectives

Given the rules for $\forall$ and $\supset$, it is easy to see that the second order definitions of the other connectives are correct; the introduction and elimination rules for these connectives in $\Box PROP_{fitch}$ are derived rules in $\Box PROP2$. We show the case for $\vee$-elim:

rule

i $\varphi \vee \psi$
$\vdots$
> j φ
> $\vdots$
> j' χ

$\vdots$
> k ψ
> $\vdots$
> k' χ

$\vdots$
χ

definition

i $\varphi \vee \psi$
$\vdots$
1. $\forall\alpha.(\varphi \supset \alpha) \supset ((\psi \supset \alpha) \supset \alpha)$ (definition i)
2. $\chi \in Prop$ (term)
3. $(\varphi \supset \chi) \supset ((\psi \supset \chi) \supset \chi)$ ($\forall$-elim 1,2)
> j φ
> $\vdots$
> j' χ

4. $\varphi \supset \chi$ ($\supset$-intro j-j')
5. $(\psi \supset \chi) \supset \chi$ ($\supset$-elim 3, 4)
> k ψ
> $\vdots$
> k' χ

6. $(\psi \supset \chi)$ ($\supset$-intro k-k')
7. χ ($\supset$-elim 5, 6)

Reiteration for assumptions only

In PTSs, contexts are ordered sequences of declarations. For these declarations an analogon of the *reiteration* rule of Fitch-style natural deduction holds: a declaration $x : A$ is derivable from any context in which it occurs. Once a declaration is added to the context, it will remain derivable on extensions of that context. Since 'complex terms' (applications and abstractions) are not recorded in PTS contexts, they cannot be reiterated directly. Given a context on which a complex term is derivable, the complex term is derivable on extensions of that context but only by 'rebuilding' it on the spot from the declarations in the context using the type derivation rules. From the point of view of natural deduction proofs the declarations in the context correspond to hypotheses. Hence in (M)PTSs reiteration holds only for hypotheses, whereas in Fitch-style natural deduction any formula may be reiterated. This discrepancy poses a problem, since we want to map the natural deduction proofs of $\Box PROP2$ to terms of $\lambda\Box PROP2$ in an inductive way (based on the natural deduction rules applied in the proofs). Therefore we propose to restrict ourselves to the following class of $\Box PROP2$-proofs in defining the mappings.

2.1.7. DEFINITION. **OK-proofs** A natural deduction proof Σ of φ in $PROP2$ is an *OK proof* if all reiterated formulas in Σ are hypotheses. Formally, for all ψ such that ψ is a premiss of reiteration in Σ, if $F(i) = \psi$, then $i = k$ for some $[k, l] \in \mathbf{H}$.

Restricting the reiteration rule to hypotheses ('h-reiteration') may seem artifical from the point of view of Fitch-deduction, but it does not mean that any 'proving power' is lost because reiteration of other formulas ('nh-reiteration') in Fitch-style deduction can be seen as a derived rule: the non-hypothesis that is to be reiterated is rebuilt 'on the spot' out of reiterated hypotheses, using the deduction rules. Hence it is clear that for any formula φ provable with a natural deduction proof Σ in $\Box PROP2$, we can find an OK-proof Σ' proving φ.

That it is not difficult to find such a proof can be seen from the following simpleminded construction that eliminates nh-reiterations from proof figures. Given a nh-reiterated formula φ, the idea is to insert a 'duplicate' of (part of) the proof of φ (the 'proof block' of φ) for the nh-reiterated occurrence of φ. Given that this inserted proof block contains only h-reiterations, 'duplication' removes the nh-reiteration. We first give an example.

2.1.8. EXAMPLE.

1.	$\mid \chi \supset \varphi$	
2.	$\mid\ \mid \chi$	
3.	$\mid\ \mid \chi \supset \varphi$	(reiteration 1)
4.	$\mid\ \mid \varphi$	
5.	$\mid\ \mid\ \mid \psi$	
6.	$\mid\ \mid\ \mid \varphi$	(reiteration 4)
7.	$\mid\ \mid \psi \supset \varphi$	
8.	$\mid \chi \supset (\psi \supset \varphi)$	
9.	$(\chi \supset \varphi) \supset (\chi \supset (\psi \supset \varphi))$	

with nh-reiteration

1.	$\mid \chi \supset \varphi$	
2.	$\mid\ \mid \chi$	
3.	$\mid\ \mid \chi \supset \varphi$	(reiteration 1)
4.	$\mid\ \mid \varphi$	
5.	$\mid\ \mid\ \mid \psi$	
6.	$\mid\ \mid\ \mid \chi$	(reiteration 2)
7.	$\mid\ \mid\ \mid \chi \supset \varphi$	(reiteration 1)
8.	$\mid\ \mid\ \mid \varphi$	
9.	$\mid\ \mid \psi \supset \varphi$	
10.	$\mid \chi \supset (\psi \supset \varphi)$	
11.	$(\chi \supset \varphi) \supset (\chi \supset (\psi \supset \varphi))$	

without nh-reiteration

The proof on the left contains an nh-reiteration of φ in line 4. It can be eliminated by substituting a duplicate of its derivation from the hypotheses χ and $(\chi \supset \varphi)$ (line 2-4) for the reiterated occurrence of φ in line 6. This results in the proof on the right were φ (line 8) is proved by $\supset$-elim from χ (line 6) and $\chi \supset \varphi$ (line 7), which are both h-reiterated.

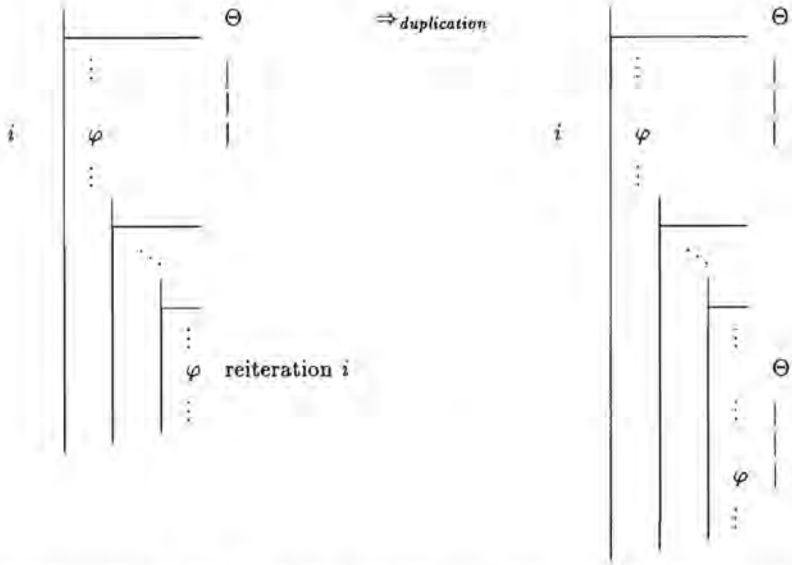
Before we can look at this elimation procedure in a general setting, we need definitions of 'proof block' and 'duplication'.

2.1.9. DEFINITION. **Proof block**
The *proof block* of a formula φ occurring at line i in a natural deduction proof Σ is the part $[k, i]$ of the interval $[k, l] \in \mathbf{I}$ in which φ lies.

Note that the interval I can be a hypothesis interval ($I \in \mathbf{H}$) as well as a modal interval ($I \in \mathbf{M}$). In both cases the proof block of φ simply consists of φ, and all formulas, notifications, and subordinate proofs that lie in the interval I 'above' φ.

2.1.10. DEFINITION. **Duplication**
Given a reiteration of a formula φ in a proof figure Σ, *duplication* is the operation that substitutes the proof block of φ for the reiterated occurrence of φ in Σ (yielding a proof Σ' of φ).

The following figure shows the general format of duplication, where the proof block of φ is indicated by Θ and the vertical lines to the right of the proof figure.



By repeated duplication all nh-reiterations can be removed from a natural deduction proof in $\Box PROP2$, since with every duplication the topmost nh-reiteration in the proof figure can be eliminated. To show this, we first make two observations.

2.1.11. OBSERVATION. By duplicating the proof block (Θ) of a reiterated formula φ, we can eliminate that reiteration of φ from the proof.

Note that if the reiterated occurrence of φ is replaced by the proof block for φ, the last rule in the derivation of φ in the duplicate block is the same as in the original proof block. Hence the application of reiteration has been eliminated from the proof, even if the last step in the derivation of φ in the proof block was itself an application of reiteration. In that case it must have been a reiteration of an occurrence of φ at an earlier line $j < i$, preceding the occurrence at line i. Therefore the derivation of φ in the copied proof block is now a reiteration of the φ in line j in stead of the φ in line i.

2.1.12. OBSERVATION. If the proof block Θ of the nh-reiterated formula φ does not contain nh-reiterations, then duplication reduces the number of nh-reiteratons in a natural deduction proof (Σ of φ) by one.

By observation 2.1.11., the particular application of nh-reiteration is eliminated. Furthermore, it is immediate that the h-reiterations present in the proof block remain h-reiterations (of the same hypotheses) in the duplicate proof block. Copying the proof block can only create one new reiteration: if φ lies in a hypothesis interval ($I \in \mathbf{H}$, as in the figure above), the hypothesis of this interval will appear in the duplicate proof block. But since this is also a case of h-reiteration, duplication does not create nh-reiterations. Hence if the proof block for φ does not contain nh-reiterations, the nh-reiteration of φ is removed from the proof by duplication.

2.1.13. PROPOSITION. *For every proof Σ of φ in $\Box PROP2$, there exists an OK-proof Σ' of the same formula in $\Box PROP2$.*

PROOF. The OK-proof Σ' is constructed from the given proof Σ, by removing all nh-reiterations by means of duplication. That this is possible follows by induction on the number of nh-reiterations in Σ. By observation 2.1.12. we can eliminate the first (topmost) nh-reiteration in Σ: since this is the *first* nh-reiteration the proof block of the nh-reiterated formula cannot contain nh-reiterations, hence we can eliminate it by duplication. Note that in the resulting proof no nh-reiterations occurs up to and including the duplicated proof block. Therefore we can again apply observation 2.1.12. to the next nh-reiteration and eliminate it by duplication. Proceeding in this way, we subsequently remove all nh-reiterations in Σ by duplication.

Since we can always find a natural deduction proof without reiterations of non-hypotheses, we can safely restrict the deductions in $\Box PROP2$ that are to be mapped to lambda terms to the class of OK-proofs.

2.2. $\Lambda\Box PROP2$

The typed λ-calculus $\Lambda\Box PROP2$ corresponding to the logic $\Box PROP2$ can be defined as in [Geuvers 1993]. Instead of a simple context divided into an 'object part' (Γ) and a 'proof part' (Δ), the modal system will have 'generalized contexts' consisting of a series of these pairs ordered by separators ($\Gamma_1; \Delta_1 \boxdot \dots \boxdot \Gamma_n; \Delta_n$), and 'generalized object contexts', a series of object parts ($\Gamma_1 \boxdot \dots \boxdot \Gamma_n$).

1 The set of *functional types* of $\Lambda\Box PROP2$, $Type^f$, is empty. The set of *predicate types* of $\Lambda\Box PROP2$, $Type^p$, consists solely of *Prop*.

2 The *object-terms* of the language of $\Lambda\Box PROP2$ form a subset of the set of *pseudoterms*, T, which is generated by the following abstract syntax.

$$T ::= Var^{Prop} | T \supset T | \forall Var^{Prop} : Prop \,.\, T | \Box T,$$

with Var^{Prop} a countable set of *object-variables*. An object-term is of a certain type only under assumption of specific types (functional or predicative) for the free variables that occur in that term. That the object term t *is of type* A if x_i is of type A_i for $1 \leq i \leq n$, is denoted by

$$x_1 : A_1, \dots, x_i : A_i \boxdot \dots \boxdot x_j : A_j, \dots, x_n : A_n \vdash t : A.$$

Here $x_1, \dots, x_n$ are different objectvariables and $A_1, \dots A_n$ are types. We call such a sequence of statements and separators a *generalized object context*. An *object context* is a sequence of declarations with different subjects that is not interrupted by separators ($x_1 : A_1, \dots, x_i : A_i$). By letting $\Gamma_1, \Gamma_2, \dots$ range over object contexts, we can represent generalized object contexts as $\Gamma_1 \boxdot \dots \boxdot \Gamma_n$. Sometimes we abbreviate even further by letting $F, F', \dots$ range over the set of generalized object contexts.

The rules for deriving judgements are the following.

$$(var)\ \frac{}{\Gamma_1 \boxdot \dots \boxdot \Gamma_n \vdash x : Prop} \quad \text{if } x : Prop \text{ in } \Gamma_n$$

$$(\supset)\ \frac{F\ \vdash\ \varphi : Prop \quad F\ \vdash\ \psi : Prop}{F\ \vdash\ \varphi \supset \psi : Prop}$$

$$(\forall)\ \frac{F, x : Prop\ \vdash\ \varphi : Prop}{F\ \vdash\ \forall x : Prop.\varphi : Prop}$$

$$(\Box)\ \frac{F\ \vdash\ \varphi : Prop}{F\ \vdash\ \Box\varphi : Prop}$$

$$(transfer)\ \frac{F\ \vdash\ \varphi : Prop}{F \boxdot \varepsilon\ \vdash\ \varphi : Prop}$$

Besides the usual rules, stated for generalized object contexts, we have the ($\Box$)-rule which allows us to modalize every formula φ, and (*transfer*) which allows us to use any non-proof term derivable on a generalized object context F on arbitrary (deep) subordinate contexts of F.

3 The set of *proof-terms* is a subset of the set of *pseudoproofs*, Pr, generated by the following abstract syntax.

$$Pr ::= Var^{pr}|PrPr|PrT|\lambda x : Type.Pr|\lambda x : T.Pr|\check{k}Pr|\check{4}Pr|\check{5}Pr|\check{b}Pr|\hat{k}Pr|\hat{d}Pr|\hat{t}Pr|C,$$

where Var^{pr} is the set of proof-variables and C is the set of constants.

That the proof term M is of type A if p_i is of type φ_i for $1 \leq i \leq l$ is denoted by

$$\Gamma_1; p_1 : \varphi_1, \ldots, p_h : \varphi_h \boxdot \ldots \boxdot \Gamma_n; p_k : \varphi_k, \ldots, p_l : \varphi_l\ \vdash\ M : A$$

where the $\Gamma_1, \ldots, \Gamma_n$ are as in 2, $p_1, \ldots p_l$ are different proof-variables and

$$\Gamma_1 \boxdot \ldots \boxdot \Gamma_n\ \vdash\ \varphi_j : Prop \quad \textit{for } k \leq j \leq l.$$

We call such a sequence of statements and separators a *generalized context*. A *proof context* is an uninterrupted sequence of declarations with different proof variables ($p_1 : \varphi_1, \ldots, p_h : \varphi_h$). By letting $\Delta_1, \Delta_2, \ldots$ range over object contexts, we can represent generalized contexts as $\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n$. We let $G, G', \ldots$ range over the set of generalized contexts.

The rules for the derivation of judgements are the following

$$(axiom)\ \frac{}{\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n\ \vdash\ p : \varphi} \quad \textit{if } p : \varphi \textit{ in } \Delta_n$$

$$(\supset\ intro)\ \frac{G, p : \varphi\ \vdash\ M : \psi}{G\ \vdash\ \lambda p : \varphi.M : \varphi \supset \psi}$$

$$(\supset\ elim)\ \frac{G\ \vdash\ M : \varphi \supset \psi \quad G\ \vdash\ N : \varphi}{G\ \vdash\ MN : \psi}$$

$$(\forall\, intro)\ \frac{\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n, x : Prop; \Delta_n\ \vdash\ M : \varphi}{\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n\ \vdash\ \lambda x : Prop.M : \forall x : Prop.\varphi} \quad \textit{if } x \notin FV(\Delta_n)$$

$$(\forall\, elim)\ \frac{\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n\ \vdash\ M : \forall x : Prop.\varphi \quad \Gamma_1 \boxdot \ldots \boxdot \Gamma_n\ \vdash\ t : Prop}{\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n\ \vdash\ Mt : \varphi[t/x]}$$

$$(K\ import)\ \frac{G\ \vdash\ M : \Box\varphi\ (: Prop)}{G \boxdot \varepsilon; \varepsilon\ \vdash\ \check{k}M : \varphi}$$

$$(4\ import)\ \frac{G\ \vdash\ M : \Box\varphi\ (: Prop)}{G \boxdot \varepsilon; \varepsilon\ \vdash\ \check{4}M : \Box\varphi}$$

$$(5\ import)\ \frac{G\ \vdash\ M : \neg\Box\varphi\ (: Prop)}{G \boxdot \varepsilon; \varepsilon\ \vdash\ \check{5}M : \neg\Box\varphi}$$

$$(B\ import)\ \frac{G\ \vdash\ M : \varphi\ (: Prop)}{G \boxdot \varepsilon; \varepsilon\ \vdash\ \check{b}M : \neg\Box\neg\varphi}$$

$$(K\ export)\ \frac{G \boxdot \varepsilon; \varepsilon\ \vdash\ M : \varphi\ (: Prop)}{G\ \vdash\ \hat{k}M : \Box\varphi}$$

$$(D\ export)\ \frac{G \boxdot \varepsilon; \varepsilon\ \vdash\ M : \varphi\ (: Prop)}{G\ \vdash\ \hat{d}M : \neg\Box\neg\varphi}$$

$$(T\ export)\ \frac{G \boxdot \varepsilon; \varepsilon\ \vdash\ M : \varphi\ (: Prop)}{G\ \vdash\ \hat{t}M : \varphi}$$

$$(Doubleneg)\ \varepsilon; \varepsilon\ \vdash\ c : \forall\alpha : Prop.((\alpha \supset \bot) \supset \bot) \supset \alpha$$

$$(Transfer_{\neg\neg})\ \frac{G\ \vdash\ c : \forall\alpha : Prop.((\alpha \supset \bot) \supset \bot) \supset \alpha}{G \boxdot \varepsilon; \varepsilon\ \vdash\ c : \forall\alpha : Prop.((\alpha \supset \bot) \supset \bot) \supset \alpha}$$

Besides the import and export rules, we need two extra rules to ensure that the double negation axioms holds on every generalized context: (*Doubleneg*) states that the axiom is derivable on an empty context, and by means of ($Transfer_{\neg\neg}$) it can be brought to any subordinate context.

We now list some meta-theoretical properties of $\Lambda\Box PROP2$ that will be of use in proving properties of the mappings. These properties hold for all MPTSs in the Modal Logic Cube, and we will prove them for all systems at once in the next chapter.

2.2.1. FACTS. Let $G\ \vdash\ M : \varphi$, where $G \equiv \Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n$ (and $F \equiv \Gamma_1 \boxdot \ldots \boxdot \Gamma_n$) be derivable in $\Lambda\Box PROP2$. We have the following properties.

1 Permutation: If for all i, $1 \leq i \leq n$, Γ'_i is a permutation of Γ_i and Δ'_i is a permutation of Δ_i, then $\Gamma'_1; \Delta'_1 \boxdot \ldots \boxdot \Gamma'_n; \Delta'_n\ \vdash\ M : \varphi$ is also derivable.

2 Substitution: If Γ_i contains $x : A$ and $\Gamma_1 \boxdot \ldots \boxdot \Gamma_i\ \vdash\ t : A$ then $\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_i/(x : A); \Delta_i[t/x] \boxdot G'[t/x]\ \vdash\ M[t/x] : \varphi[t/x]$ is also derivable, where $G'[t/x] \equiv \Gamma_{i+1}[t/x]; \Delta_{i+1}[t/x]\ \ldots \boxdot \Gamma_n[t/x]; \Delta_n[t/x]$.

3 Thinning for generalized object contexts: If for all i, $1 \leq i \leq n$, $\Gamma'_i \supseteq \Gamma_i$ with Γ'_i object contexts and all object-variables occurring as subjects in $\Gamma'_1 \boxdot \ldots \boxdot \Gamma'_n$ are different, then $\Gamma'_1 \boxdot \ldots \boxdot \Gamma'_n\ \vdash\ M : \varphi$ is also derivable.

Thinning for generalized contexts: If for all i, $1 \leq i \leq n$, $\Gamma'_i \supseteq \Gamma_i$ with Γ'_i object contexts and $\Delta'_i \supseteq \Delta_i$, Δ'_i proof-contexts and all object and proof variables occurring as subjects in $\Gamma'_1; \Delta'_1 \boxdot \ldots \boxdot \Gamma'_n; \Delta'_n$ are different, then $\Gamma'_1; \Delta'_1 \boxdot \ldots \boxdot \Gamma'_n; \Delta'_n\ \vdash\ M : \varphi$ is also derivable.

Note that for a generalized object context F, the combination of Thinning with the *transfer* rule makes it possible to derive any term $\varphi : Prop$ on F from the fact that it is derivable on a 'prefix' of $F (\equiv \Gamma'_1 \boxdot \ldots \boxdot \Gamma'_i$ where $1 \leq i \leq n$ and $\Gamma'_j \subseteq \Gamma_j$ for $1 \leq j \leq i)$. We call this *Strong Thinning*.

4 Closure or Subject-reduction: If $M \twoheadrightarrow_\beta M'$, then $G \vdash M' : \varphi$ is also derivable.

5 Stripping or Generation:

$\Gamma_1 \boxdot \ldots \boxdot \Gamma_n \vdash x : Prop$ *(x an object variable)* $\Rightarrow$
$x : Prop \in \Gamma_i$ *for some* $i,\ 1 \leq i \leq n$

$F \vdash \varphi \supset \psi : Prop \Rightarrow F \vdash \varphi : Prop$ *and* $F \vdash \psi : Prop$

$\Gamma_1 \boxdot \ldots \boxdot \Gamma_n \vdash \forall x : Prop.\psi : Prop \Rightarrow$
$\Gamma_1 \boxdot \ldots \boxdot \Gamma_n, x : Prop \vdash \psi : Prop$

$F \vdash \Box\varphi : Prop \Rightarrow F \vdash \varphi : Prop$

$\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n \vdash p : \varphi$ *(p a proof variable)* $\Rightarrow p : \varphi \in \Delta_n$

$\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n \vdash \lambda x : Prop.M : \varphi \Rightarrow$
$\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n, x : Prop; \Delta_n \vdash M : \psi$
with $\varphi \equiv \forall x : Prop.\psi$ *for some* ψ

$\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n \vdash \lambda p : \chi.M : \varphi$ *(χ a proposition)* $\Rightarrow$
$\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n, p : \chi \vdash M : \psi$ *with* $\varphi \equiv \chi \supset \psi$ *for some* ψ

$G \vdash MN : \varphi$ *(N a proof)* $\Rightarrow G \vdash M : \psi \supset \chi$ *and* $G \vdash N : \psi$
with $\varphi \equiv \chi$ *for some* ψ, χ

$\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n \vdash Mt : \varphi$ *(t an object)* $\Rightarrow$
$\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n \vdash M : \forall x : Prop.\psi$, *and* $\Gamma_1 \boxdot \ldots \boxdot \Gamma_n \vdash t : Prop$
with $\varphi \equiv \psi[t/x]$, *for some* ψ, φ

$G \vdash \check{k}M : \varphi$ *(M a proof)* $\Rightarrow G \equiv G' \boxdot \Gamma; \Delta$ *and* $G' \vdash M : \Box\psi$ *where* $\psi \equiv \varphi$

$G \vdash \check{4}M : \varphi$ *(M a proof)* $\Rightarrow G \equiv G' \boxdot \Gamma; \Delta$ *and* $G' \vdash M : \Box\psi$ *where* $\Box\psi \equiv \varphi$

$G \vdash \check{5}M : \varphi$ *(M a proof)* $\Rightarrow G \equiv G' \boxdot \Gamma; \Delta$ *and* $G' \vdash M : \neg\Box\psi$ *where* $\neg\Box\psi \equiv \varphi$

$G \vdash \check{b}M : \varphi$ *(M a proof)* $\Rightarrow G \equiv G' \boxdot \Gamma; \Delta$ *and* $G' \vdash M : \psi$ *where* $\neg\Box\neg\psi \equiv \varphi$

$G \vdash \hat{k}M : \varphi$ *(M a proof)* $\Rightarrow G \boxdot \varepsilon \vdash M : \psi$ *where* $\Box\psi \equiv \varphi$

$G \vdash \hat{d}M : \varphi$ *(M a proof)* $\Rightarrow G \boxdot \varepsilon \vdash M : \psi$ *where* $\neg\Box\neg\psi \equiv \varphi$

$G \vdash \hat{t}M : \varphi$ *(M a proof)* $\Rightarrow G \boxdot \varepsilon \vdash M : \psi$ *where* $\psi \equiv \varphi$

$G \vdash c : \varphi \Rightarrow \varphi \equiv \forall \alpha : Prop.((\alpha \supset \bot) \supset \bot) \supset \alpha$

2.3. Mapping □*PROP*2 to Λ□*PROP*2

To an OK-deduction of $\varphi_1, \ldots, \varphi_n \vdash \psi$ in $\Box PROP2$ we are going to associate an object-context Γ and a proof-term M such that $\Gamma; p_1 : \varphi_1, \ldots, p_n : \varphi_n \vdash M : \psi$.

In order to make M a faithful representation of the deduction in $\Box PROP2$, Γ should assign types to all the free term-variables in the deduction that are not 'bound by a ∀' at any later stage.

2.3.1. DEFINITION. **Term-contexts**
For every term t of the language of $\Box PROP2$ we define a context Γ_t such that $\Gamma_t \vdash t : D$ (in $\Lambda\Box PROP2$) if $t \in D$ (in $\Box PROP2$), as follows

$$\begin{array}{rcl} t \equiv x^{Prop} & \Rightarrow & \Gamma_t := x^{Prop} : Prop, \\ t \equiv \varphi \supset \psi & \Rightarrow & \Gamma_t := \Gamma_\varphi \cup \Gamma_\psi \\ t \equiv \forall x \in Prop.\varphi & \Rightarrow & \Gamma_t := \Gamma_\varphi / (x : Prop) \\ t \equiv \Box\varphi & \Rightarrow & \Gamma_t := \Gamma_\varphi \end{array}$$

This definition is correct in the sense that every term t is derivable on 'its' context:

2.3.2. LEMMA. **Term-context lemma**
For all terms $t \in Prop$ of $\Box PROP2$, $\Gamma_t \vdash t : Prop$ in $\Lambda\Box PROP2$.

PROOF. By induction on the structure of the term.

The mapping does not use the annotation of the proof figures, but some indexing is needed to link the hypotheses to variables in $\Lambda\Box PROP2$.

2.3.3. CONVENTION. **Indexing**

Hypotheses (propositions as well as variables) are numbered consecutively, reading the proof figure top to bottom

Reiterated formulas receive the index of the hypothesis they are a reiteration of

In this way all formulas with a trivial proof (hypotheses and reiterations of hypotheses) will have a unique index as will the proof- and object-variables assigned to them by the mapping.

The OK-deductions in $\Box PROP2$ are mapped to a term/object-context (Γ), proof-context (Δ) and a term (M) in $\Lambda\Box PROP2$ by an inductively defined mapping, '!'. This mapping constructs a $\Lambda\Box PROP2$-term 'from the inside out': starting from the conclusion it inductively works its way upward through the proof figure guided by the last applied deduction rule until a formula with an 'atomic' proof is reached. In the OK-proofs these formulas without further justification are (reiterations of) assumptions, terms and axioms. During the deconstruction each !-step generates a corresponding typetheoretical 'inference rule', for which we have enough information to determine the types. The contexts and proof terms in these inference rules can only be determined after the formulas with atomic proofs are reached. ! maps these directly to object- and proof-variables and starting from these 'atomic' proof terms we move back down the chain of inference rules to obtain the $\Lambda\Box PROP2$-term corresponding to the complete proof figure.

The definition of '!' has a clause for each deduction rule of $\Box PROP2$, and in addition a clause to deal with derivations with hypotheses. In these derivations some assumptions are not discharged and '!' has to map these to the proof context. In the case for $\supset$-elim contexts are joined by means of a special union operation '$\cup_{zip}$', in other cases the union '$\cup^*$' is used. These operations will be explained later, along with the underlinings of parts of the context that are introduced in some of the other cases. Finally, we assume that in an OK-deduction all bound variables are chosen to be different and in such a way that they differ from the free ones.

2.3.4. DEFINITION. **'!', the mapping**

prop-assumption

$$\psi^i \quad \Rightarrow \quad \underline{\Gamma_\psi}; p_i : \psi \ \vdash \ p_i : \psi$$

var-assumption

$$x \in Prop^i \quad \Rightarrow \quad x_i : Prop\ \varepsilon \ \vdash \ x_i : Prop$$

term

$$\frac{}{t \in Prop}\ \text{term} \quad \Rightarrow \quad \underline{\Gamma_t} \ \vdash \ t : Prop$$

axiom

$$\frac{}{\forall \alpha \in Prop.((\alpha \supset \bot) \supset \bot) \supset \alpha}\ \text{axiom} \quad \Rightarrow \quad \varepsilon; \varepsilon \ \vdash \ c : \forall \alpha : Prop.((\alpha \supset \bot) \supset \bot) \supset \alpha$$

⊃-intro

$$\begin{array}{l|l} \varphi^i & \Sigma \\ \hline \vdots & | \\ \psi & | \end{array}$$

$$\varphi \supset \psi$$

$$\Rightarrow$$

$$\frac{\Gamma_1; \Delta_1 \boxtimes \ldots \boxtimes \Gamma_n; \Delta_n \ \vdash \ \Sigma^! : \psi}{\Gamma_1 \cup^* \underline{\Gamma_\varphi}; \Delta_1 \boxtimes \ldots \boxtimes \Gamma_n; \Delta_n/(p_i : \varphi) \ \vdash \ \lambda p_i : \varphi.\Sigma^! : \varphi \supset \psi}$$

⊃-elim

$$\begin{array}{lll} & \Sigma & \Theta \\ \vdots & | & | \\ \varphi \supset \psi & | & | \\ \vdots & & | \\ \varphi & & | \\ \psi & & \end{array}$$

$$\Rightarrow$$

$$\frac{G_1 \ \vdash \ \Sigma^! : \varphi \supset \psi \qquad G_2 \ \vdash \ \Theta^! : \varphi}{G_1 \cup_{zip} G_2 \ \vdash \ \Sigma^! \Theta^! : \psi}$$

∀-intro

$$\begin{array}{|l} x \in Prop' \\ \hline \vdots \\ \psi \end{array} \quad \Sigma$$

$\forall x \in Prop.\psi$

$\Rightarrow$

$$\frac{\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n \ \vdash \ \Sigma' : \psi}{\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n/(x_i : Prop); \Delta_n \ \vdash \ \lambda x_i : Prop.\Sigma' : \forall x : Prop.\psi}$$

∀-elim

$$\begin{array}{l} \vdots \\ \forall x \in Prop.\psi \end{array} \quad \Sigma$$

$\vdots$

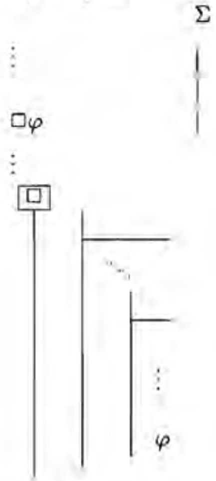
$t \in Prop$

$\psi[t/x]$

$\Rightarrow$

$$\frac{\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n \ \vdash \ \Sigma' : \forall x : Prop.\psi \qquad \Gamma_t \ \vdash \ t : Prop}{\Gamma_1 \cup^* \Gamma_t; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n \ \vdash \ \Sigma' t : \psi[t/x]}$$

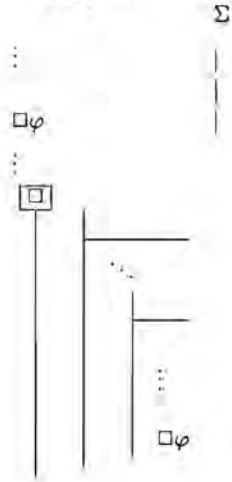
K-import



$$\Rightarrow$$

$$\frac{G \vdash \Sigma^! : \Box\varphi}{G \boxed{\Box}\, \varepsilon;\varepsilon \vdash \check{k}(\Sigma^!) : \varphi}$$

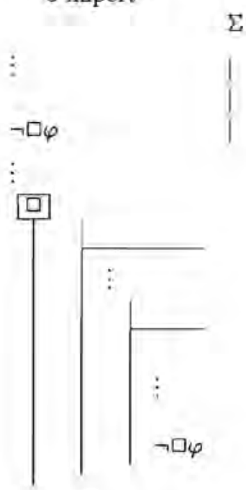
4-import



$$\Rightarrow$$

$$\frac{G \vdash \Sigma^! : \Box\varphi}{G \boxed{\Box}\, \varepsilon;\varepsilon \vdash \check{4}(\Sigma^!) : \Box\varphi}$$

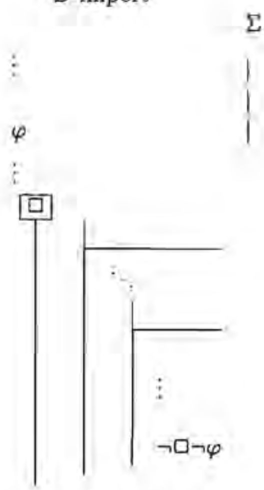
5-import



$$\Rightarrow$$

$$\frac{G \vdash \Sigma^! : \neg\Box\varphi}{G \boxed{\Box}\, \varepsilon;\varepsilon \vdash \check{5}(\Sigma^!) : \neg\Box\varphi}$$

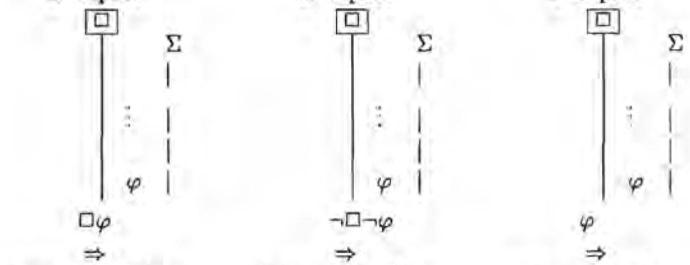
B-import



$$\Rightarrow$$

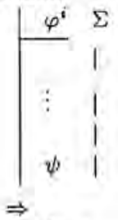
$$\frac{G \vdash \Sigma^! : \varphi}{G \boxed{\Box}\, \varepsilon;\varepsilon \vdash \check{b}(\Sigma^!) : \neg\Box\neg\varphi}$$

K-export *D*-export *T*-export



$$\frac{G \boxdot \varepsilon;\varepsilon \vdash \Sigma^{!} : \varphi}{G \vdash \hat{k}(\Sigma^{!}) : \Box\varphi} \qquad \frac{G \boxdot \varepsilon;\varepsilon \vdash \Sigma^{!} : \varphi}{G \vdash \hat{d}(\Sigma^{!}) : \neg\Box\neg\varphi} \qquad \frac{G \boxdot \varepsilon;\varepsilon \vdash \Sigma^{!} : \varphi}{G \vdash \hat{t}(\Sigma^{!}) : \varphi}$$

hypothesis



$$\frac{\Gamma;\Delta \vdash \Sigma^{!} : \psi}{\Gamma \cup \Gamma_{\varphi};\Delta \cup p_i : \varphi \vdash \Sigma^{!} : \psi}$$

If for Σ a deduction in *PROP*2, $\Gamma;\Delta \vdash \Sigma^{!} : \varphi$ is the judgement that we obtain from Σ by the definition above, we write Γ_Σ for Γ and Δ_Σ for Δ.

2.3.5. FACTS.

1 For Σ an OK-deduction in $\Box$*PROP*2 there is a one-to-one-correspondence between occurrences of non-discharged formulas of Σ and declarations of variables to the same formula in Δ_Σ.

2 In the case for the $\forall$-intro rule the variable x does not occur free in the proof-context Δ.

3 During every stage of applying '!' to an OK-deduction the resulting context is a generalized context (all statements in the context have a different object- or proof-variable as subject).

For applications of $\supset$-elim in the proof figure the mapping has to 'unite' two generalized contexts. For non-modal propositional logic we could simply take the union of the object-contexts ($\Gamma_1 \cup \Gamma_2$) and the proof-contexts ($\Delta_1 \cup \Delta_2$). However, in modal type has to deal with two generalized contexts *which may be of different modal depth* (contain a different number of $\boxdot$s). This consideration leads to an operation called *zip-union*, which can be understood as 'zipping' two generalized contexts together.

Given generalized contexts $G_1 \equiv \Gamma_1;\Delta_1 \boxdot \ldots \boxdot \Gamma_n;\Delta_n$ and $G_2 \equiv \Gamma'_1;\Delta'_1 \boxdot \ldots \boxdot \Gamma'_m;\Delta'_m$ (where $n \geq m$), the idea is to first line up the contexts in such a way that their '$\boxdot$s' are (vertically) aligned starting from their rightmost $\Gamma;\Delta$-part:

$$\begin{array}{rl} \Gamma_1;\Delta_1 \boxdot \ldots \boxdot \Gamma_{n-m};\Delta_{n-m} \boxdot \ldots & \boxdot \;\; \Gamma_n;\Delta_n \\ \Gamma'_1;\Delta'_1 \boxdot \ldots & \boxdot \;\; \Gamma'_m;\Delta'_m \end{array}$$

The 'zipper' then moves from right to left, joining the corresponding $\Gamma;\Delta$-parts into new $\Gamma;\Delta$-parts by taking the union of the object-contexts and the proof-contexts:

$$\left.\begin{array}{cl} \Gamma_1;\Delta_1 \boxdot \ldots \boxdot \Gamma_{n-m};\Delta_{n-m} & \boxdot \;\ldots \\ \Gamma'_1;\Delta'_1 & \boxdot \;\ldots \end{array}\right\} \boxdot \Gamma_{n-j}\cup\Gamma'_{m-j};\Delta_{n-j}\cup\Delta'_{m-j} \boxdot \ldots \boxdot \Gamma_n\cup\Gamma'_m;\Delta_n\cup\Delta'_m$$

This 'zipping' continues further to the left until the leftmost $\Gamma;\Delta$-part of the shortest context is reached. The proof-context Δ'_1 is joined with the corresponding proof-context of G_1, Δ_{n-m}. The object-context Γ'_1 is treated differently. Statements in Γ'_1 may have been underlined during the application of the mapping '!'. The set of these statements, $\underline{\Gamma'_1}$, contains all terms $(\varphi : Prop)$ that do not function as the hypothesis of a $\forall$-subordinate proof in the part of the proof figure that is mapped to G_2. This well-typedness information could be 'in the way' during the rest of the proof (see example below). Therefore it is stored in the leftmost object context (Γ_1) of the longest generalized context participating in the zip-union (G_1). To avoid 'double declarations', we do not add these well-typedness statements to Γ_1 by '$\cup$' but by '$\cup^*$', i.e. we add only those statements in $\underline{\Gamma'_1}$ that do not already occur (anywhere) in G_1. The remainder of Γ'_1, $\Gamma'_1/\underline{\Gamma'_1}$, is joined to Γ_{n-m}.

2.3.6. DEFINITION. **Zipping**
For two generalized contexts G_1 and G_2 (assuming without loss of generality that $G_1 \equiv \Gamma_1;\Delta_1 \boxdot \ldots \boxdot \Gamma_n;\Delta_n$ and $G_2 \equiv \Gamma'_1;\Delta'_1 \boxdot \ldots \boxdot \Gamma'_m;\Delta'_m$, where $n \geq m$)

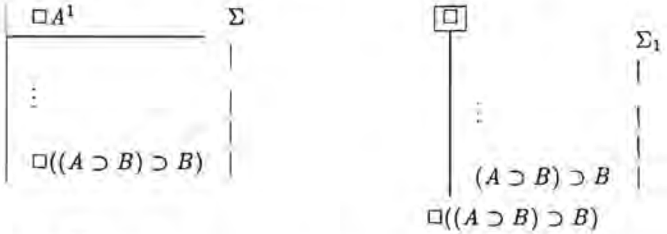
$$G_1 \cup_{zip} G_2 \equiv \Gamma_1 \cup^* \underline{\Gamma'_1};\Delta_1 \boxdot \ldots \boxdot \Gamma_{n-m} \cup \Gamma'_1/(\underline{\Gamma'_1});\Delta_{n-m} \cup \Delta'_1 \boxdot \ldots \boxdot \Gamma_n \cup \Gamma'_m;\Delta_n \cup \Delta'_m,$$

where $\underline{\Gamma'_1}$ is the set of all statements in Γ'_1 that have been underlined by '!'.

2.3.7. EXAMPLE. We illustrate how '!' constructs a proof term from a modal natural deduction proof by means of a simple example.

1.	$\Box A$		1.	$\Box A^1$
	$\boxed{\Box}$			$\boxed{\Box}$
2.	$A \supset B$		2.	$A \supset B^2$
3.	A	(K-import 1)	3.	A
4.	B	$\supset$-elim 2,3	4.	B
5.	$(A \supset B) \supset B$	($\supset$-intro 2-4)	5.	$(A \supset B) \supset B$
6.	$\Box((A \supset B) \supset B)$	(K-export,5)	6.	$\Box((A \supset B) \supset B)$

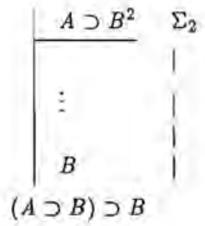
The proof figure on the left is a derivation with hypotheses in $\Box PROP2$. Indexing it according to the convention given earlier yields the figure on the right to which '!' can be applied.



$$\Rightarrow \qquad\qquad \Rightarrow$$

$$\frac{\Gamma;\Delta \vdash \Sigma^{!} : \Box((A \supset B) \supset B)}{\Gamma;\Delta \cup p_1 : \Box A \vdash \Sigma^{!} : \Box((A \supset B) \supset B)}\ (7) \qquad \frac{G \boxdot \varepsilon;\varepsilon \vdash \Sigma_1^{!} : (A \supset B) \supset B}{G \vdash \hat{k}(\Sigma_1^{!}) : \Box((A \supset B) \supset B)}\ (6)$$

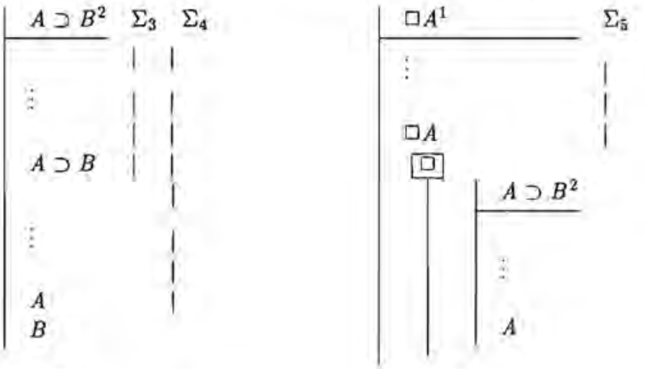
The general form of the deduction proof is that of a derivation with the hypothesis $\Box A$. Applying the hypothesis clause maps this undischarged assumption to a proof variable that is added to the proof context. The 'breakdown' of the proof is started by reducing it to a proof Σ_1 of $(A \supset B) \supset B$ in a modal subordinate proof, and subsequently to a proof Σ_2 of B under the additional assumption $A \supset B$.



$$\Rightarrow$$

$$\frac{\Gamma_1;\Delta_1 \boxdot \ldots \boxdot \Gamma_n;\Delta_n \vdash \Sigma_2^{!} : B}{\Gamma_1 \cup \underline{\Gamma_{A \supset B}};\Delta_1 \boxdot \ldots \boxdot \Gamma_n;\Delta_n/(p_2 : A \supset B) \vdash \lambda p_2 : A \supset B.\Sigma_2^{!} : (A \supset B) \supset B}\ (5)$$

The proof of B is reduced to two proofs: Σ_3 of $A \supset B$ and Σ_4 of A. Proving the imported formula A in the modal subordinate is then reduced to proving $\Box A$, by Σ_5 in the main proof.



$$\Rightarrow \qquad\qquad \Rightarrow$$

$$\frac{G_1 \vdash \Sigma_3^{!} : A \supset B \quad G_2 \vdash \Sigma_4^{!} : A}{G_1 \cup_{zip} G_2 \vdash \Sigma_3^{!}\Sigma_4^{!} : B}\ (4) \qquad \frac{G \vdash \Sigma_5^{!} : \Box A}{G \boxdot \varepsilon;\varepsilon \vdash \hat{k}(\Sigma_5^{!}) : \Box A}\ (2)$$

$A \supset B$ and $\Box A$ have atomic proofs. They are both assumptions of the natural deduction proof, and consequently '!' maps them to proof variables.

$$\Sigma_3 = A \supset B^2 \qquad\qquad \Sigma_5 = \Box A^1$$
$$\Rightarrow \qquad\qquad \Rightarrow$$
$$\Gamma_{A\supset B}; p_2 : A \supset B \vdash p_2 : A \supset B \ (3) \quad \Gamma_{\Box A}; p_1 : \Box A \vdash p_1 : \Box A \ (1)$$

From these first type theoretical statements, we move back through the clauses and obtain a term for the proof figure:

1. $\underline{\Gamma_{\Box A}}; p_1 : \Box A \vdash p_1 : \Box A$
2. $\underline{\Gamma_{\Box A}}; p_1 : \Box A \boxdot \varepsilon; \varepsilon \vdash \check{k}p_1 : A$
3. $\underline{\Gamma_{A\supset B}}; p_2 : A \supset B \vdash p_2 : A \supset B$
4. $\underline{\Gamma_{\Box A}} \cup \underline{\Gamma_{A\supset B}}; p_1 : \Box A \boxdot \varepsilon; p_2 : A \supset B \vdash p_2(\check{k}p_1) : B$
5. $\underline{\Gamma_{\Box A}} \cup \underline{\Gamma_{A\supset B}}; p_1 : \Box A \boxdot \varepsilon; \varepsilon \vdash \lambda p_2 : (A \supset B).p_2(\check{k}p_1) : (A \supset B) \supset B$
6. $\underline{\Gamma_{\Box A}} \cup \underline{\Gamma_{A\supset B}}; p_1 : \Box A \vdash \hat{k}(\lambda p_2 : (A \supset B).p_2(\check{k}p_1)) : \Box((A \supset B) \supset B)$
7. $\underline{\Gamma_{\Box A}} \cup \underline{\Gamma_{A\supset B}}; p_1 : \Box A \vdash \hat{k}(\lambda p_2 : (A \supset B).p_2(\check{k}p_1)) : \Box((A \supset B) \supset B)$

In the application of inference rule 4 (line 4), contexts of different modal depth are 'zipped' together. The term $\check{k}p_1$ proves the imported version A of the *global assumption* $\Box A$ on the generalized context $\underline{\Gamma_{\Box A}}; p_1 : \Box A \boxdot \varepsilon; \varepsilon$, whereas p_2 proves the *local assumption* $A \supset B$ on the 'simple' context $\Gamma_{A\supset B}; p_2 : A \supset B$. Zipping these contexts together (starting from the right) yields a generalized context in which the 'scope' of the assumptions is correctly maintained: the proof variable p_2 which is local to the subordinate context ends up to the right of the '$\boxdot$'.

The object context $\Gamma_{A\supset B}$ has been underlined by the mapping and is hence referred to the leftmost object context, leaving an empty subordinate object context. This is important, since after the discharge of the assumption $A \supset B$ (5) the subordinate context is completely empty $(\varepsilon; \varepsilon)$, which makes it legitimate to apply the export rule to obtain the modal conclusion.

Note that the final (hypothesis) rule, 7, makes no difference: the statement $p_2 : A \supset B$ is already presented in the object context. The rule only alters the context when it adds a statement corresponding to an hypothesis that is not used in the derivation.

In the proof of soundness of '!' we need the property that typability is preserved under 'zipping': if a proof term is derivable on a generalized context G, it is derivable on all generalized contexts obtained by 'zipping' G with another generalized context.

2.3.8. LEMMA. **Zip-lemma for proofs**
Assuming (without loss of generality) that $G_1 \equiv \Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n$ *and* $G_2 \equiv \Gamma'_1; \Delta'_1 \boxdot \ldots \boxdot \Gamma'_m; \Delta'_m$, *where* $n \geq m$, *we have for all generalized contexts* G_1, G_2 *such that* $G_1 \cup_{zip} G_2$ *is a generalized context, and types* φ:

i If $G_1 \vdash M : \varphi$, *then* $G_1 \cup_{zip} G_2 \vdash M : \varphi$

ii If $G_2 \vdash M : \varphi$, *then* $G_1 \cup_{zip} G_2 \vdash M : \varphi$

PROOF. i is proved by Thinning, and ii by induction on the structure of the term M.

Proof of i. Suppose that that (1) $G_1 \vdash \varphi : Prop$, we have to show that $G_1 \cup_{zip} G_2 \vdash \varphi : Prop$, i.e. $\Gamma_1 \cup^* \underline{\Gamma_1'}; \Delta_1 \boxdot \ldots \boxdot \Gamma_{n-m} \cup \Gamma_1'/(\underline{\Gamma_1'}); \Delta_{n-m} \cup \Delta_1' \boxdot \ldots \boxdot \Gamma_n \cup \Gamma_m'; \Delta_n \cup \Delta_m' \vdash \varphi : Prop$. This follows from (1) by Thinning, since $G_1 \cup_{zip} G_2$ is a generalized context and:

- $\Gamma_1 \cup^* \underline{\Gamma_1'} \supseteq \Gamma_1$, $\Gamma_{n-m} \cup \Gamma_1'/(\underline{\Gamma_1'}) \supseteq \Gamma_{n-m}$, and $\Gamma_{(n-m)+i} \cup \Gamma_i' \supseteq \Gamma_i$ for all $i :\ 1 \leq i \leq m$
- $\Delta_{(n-m)+i} \cup \Delta_i' \supseteq \Delta_i$ for all $i :\ 1 \leq i \leq m$.

Proof of ii. By induction on the structure of the term (M), for all generalized contexts G_1, G_2 such that $G_1 \cup_{zip} G_2$ is a generalized context, and types φ. This quantification over generalized contexts is needed in the modal cases, as can be seen from the case for K-import:

$M \equiv \check{k} M_1$ is an import term. If $\Gamma_1'; \Delta_1' \boxdot \ldots \boxdot \Gamma_m'; \Delta_m' \vdash \check{k} M_1 : \varphi$, then by Stripping $\Gamma_1'; \Delta_1' \boxdot \ldots \boxdot \Gamma_{m-1}'; \Delta_{m-1}' \vdash M_1 : \Box\varphi$. Then by IH $\Gamma_1 \cup^* \underline{\Gamma_1'}; \Delta_1 \boxdot \ldots \boxdot \Gamma_{n-m} \cup \Gamma_1'/(\underline{\Gamma_1'}); \Delta_{n-m} \cup \Delta_1' \boxdot \ldots \boxdot \Gamma_{n-1} \cup \Gamma_{m-1}'; \Delta_{n-1} \cup \Delta_{m-1}' \vdash M_1 : \Box\varphi$. This is taking the zip-union of $G_1/(\boxdot \Gamma_n; \Delta_n)$ and $G_2/(\boxdot \Gamma_m'; \Delta_m')$, which is allowed since the induction on the structure of the term quantifies over all pairs generalized contexts whose zip-union is again a generalized context (that $G_1/(\boxdot \Gamma_n; \Delta_n) \cup_{zip} G_2/(\boxdot \Gamma_m'; \Delta_m')$ is a generalized context follows from the assumption that $G_1 \cup_{zip} G_2$ is a generalized context). Hence by K-import
$\Gamma_1 \cup^* \underline{\Gamma_1'}; \Delta_1 \boxdot \ldots \boxdot \Gamma_{n-m} \cup \Gamma_1'/(\underline{\Gamma_1'}); \Delta_{n-m} \cup \Delta_1' \boxdot \ldots \boxdot \Gamma_{n-1} \cup \Gamma_{m-1}'; \Delta_{n-1} \cup \Delta_{m-1}' \boxdot \varepsilon; \varepsilon \vdash \check{k} M_1 : \varphi$,
and so by Thinning $(\Gamma_n \cup \Gamma_m' \supseteq \varepsilon,\ \Delta_n \cup \Delta_m' \supseteq \varepsilon)$
$\Gamma_1 \cup^* \underline{\Gamma_1'}; \Delta_1 \boxdot \ldots \boxdot \Gamma_{n-m} \cup \Gamma_1'/(\underline{\Gamma_1'}); \Delta_{n-m} \cup \Delta_1' \boxdot \ldots \boxdot \Gamma_{n-1} \cup \Gamma_{m-1}'; \Delta_{n-1} \cup \Delta_{m-1}' \boxdot \Gamma_n \cup \Gamma_m'; \Delta_n \cup \Gamma_m' \vdash \check{k} M_1 : \varphi$, in other words $G_1 \cup_{zip} G_2 \vdash \check{k} M : \varphi$.

In the export cases we use the 'inductive freedom' in a similar way, by applying the IH to the zip-union of $G_1 \boxdot \varepsilon; \varepsilon$ and $G_2 \boxdot \varepsilon; \varepsilon$.

Now we are ready to prove the soundness of '!'. Note that the definition of derivation requires that all modal subproofs in Σ have been closed, hence the 'n' of the context G_Σ generated by mapping Σ has to be equal to 1.

2.3.9. THEOREM. *If Σ is an OK natural deduction proof of φ in $\Box PROP2$, then $\Gamma_\Sigma; \Delta_\Sigma \vdash \Sigma^! : \varphi$ is derivable in $\Lambda\Box PROP2$.*

PROOF. By induction on the deduction Σ.

The cases for object variable, axiom, term and the modal cases are immediate by the rules of $\Lambda\Box PROP2$ and the Term-context lemma. The cases for reiteration, assumption and $\forall$-elim require Thinning. Given the Zip-lemma, the $\supset$-elim case is straightforward. For $\supset$-intro and $\forall$-intro we need Permutation, we show the case for $\supset$-intro:

$$\Rightarrow \frac{\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n \vdash \Sigma^! : \psi}{\Gamma_1 \cup \underline{\Gamma_\varphi}; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n/(p_i : \varphi) \vdash \lambda p_i : \varphi.\Sigma^! : \varphi \supset \psi}$$

By IH

$$\Rightarrow \Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n \vdash \Sigma^! : \psi \ (1)$$

We have to prove $\Gamma_1 \cup^* \Gamma_\varphi; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n/(p_i : \varphi) \vdash \lambda p_i : \varphi.\Sigma^! : \varphi \supset \psi \quad (1')$.
Given the $\supset$-I rule of $\Lambda\Box PROP2$ it is sufficient to prove that
$\Gamma_1 \cup^* \Gamma_\varphi; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta'_n, p_i : \varphi \vdash \Sigma^! : \psi$.

Since $\Gamma_1 \cup^* \Gamma_\varphi \supseteq \Gamma_1$, by Thinning we have $\Gamma_1 \cup^* \Gamma_\varphi; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n \vdash \Sigma^! : \psi$. What remains to be shown is that $\Sigma^! : \psi$ is derivable on this context with $\Delta'_n, p_i : \varphi \equiv \Delta_n$, i. e. on a proof-context where $p_i : \varphi$ is in rightmost position.

There are two possible cases:

1 $p_i : \varphi$ is already an element of Δ_n. Then by Permutation we can shuffle Δ_n in such a way that it ends up in rightmost position: $\Gamma_1 \cup^* \Gamma_\varphi; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta'_n, p_i : \varphi \vdash \Sigma^! : \psi$.

2 $p_i : \varphi$ is not in Δ_n. Then by Thinning, $\Delta'_n \equiv (\Delta_n \cup p_i : \varphi) \supseteq \Delta_n$
$\Gamma_1 \cup^* \Gamma_\varphi; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n \cup p_i : \varphi \vdash \Sigma^! : \psi$. Using Permutation as in 1, $p_i : \varphi$ can then be brought in rightmost position.

In both cases we end up with $\Gamma_1 \cup^* \Gamma_\varphi; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta'_n, p_i : \varphi \vdash \Sigma^! : \psi$, and so
$\Gamma_1 \cup^* \Gamma_\varphi; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n/(p_i : \varphi) \vdash \lambda p_i : \varphi.\Sigma^! : \varphi \supset \psi$.

2.4. Mapping $\Lambda\Box PROP2$ to $\Box PROP2$

In this section we define a mapping,'?', that maps lambda-terms in $\Lambda\Box PROP2$ to natural deduction proofs in $\Box PROP2$. Contrary to '!' that builds a term for a proof figure 'from the inside out', '?' builds a proof figure reading the term 'from the outside in'. During the deconstruction of the lambda term each '?'-step generates a part of the proof figure until a variable or a constant is reached, corresponding to an item in the proof figure with an 'atomic' derivation: a hypothesis (or reiteration thereof), an axiom or a notification.

According to the definitions of derivation and derivation with hypotheses for $\Box PROP2$ (section 2.1), a proof figure in which the conclusion is an element of a modal interval is not a proper natural deduction proof. Hence '?' should yield a derivation $M^?$ of φ in $\Box PROP2$ for those proof terms M that are derivable on a simple context: $\Gamma; \Delta \vdash M : \varphi$. Proof terms derivable on a generalized context $\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n$ ($n \geq 2$) can only represent part of a derivation, since the $\boxdot$'s in the context indicate that at least one modal interval in the corresponding proof figure is left 'open'. Still we have to define '?' for generalized contexts G rather than simple contexts $\Gamma; \Delta$, since processing the modal steps represented in M may force us to consider subordinate contexts of $\Gamma; \Delta$.

Consequently '?' can be applied to any derivable proof term, and the result of its application need not be a derivation. As noted above, it may only be a 'pre-derivation': all items occurring in it are hypotheses or the result of applying a deduction rule to preceding items, but the conclusion could be an element of a modal interval. In general applying '?' will result in a proof figure $M^?$ which is a pre-derivation in combination with the constellation of hypothesis intervals and modal intervals represented in the context (G) on which M is derivable. $M^?$ may contain reiterations or imports of hypotheses that do not appear in the proof figure, but are represented in the context.

2.4.1. DEFINITION. **Pre-derivation with respect to G**
A *pre-derivation of C with respect to G* is a proof figure $\mathbf{D}$ with interval $D = [1, n]$ and formulas and notifications $F_1, \ldots F_n$ that satisfies the following conditions.

1 $F_n = C$

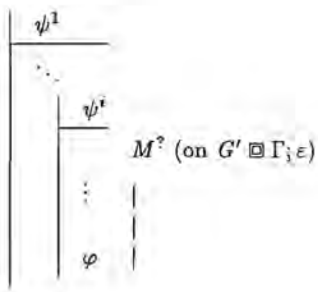
2 Every formula or notification $F_i (1 \leq i \leq n)$ is a hypothesis or the result of the application of a deduction rule on a number of formulas or notifications preceding F_i or represented in the generalized context G.

In a generalized context G the proof contexts $\Delta_i (1 \leq i \leq n)$ each represent a group of nested hypothesis intervals with formulas as hypotheses, the object contexts $\Gamma_i (1 \leq i \leq n)$ each represent a group of nested hypothesis intervals with notifications as hypotheses, and the $\boxdot$'s each represent a modal interval.

Even for proof terms derivable on a simple context we have to take into account that part of the derivation may be represented in the context. When the proof context is empty, $\Gamma; \varepsilon \vdash M : \varphi$, applying '?' will result in a derivation without hypotheses of $\Box PROP2$. When the proof context is non-empty, $\Gamma; \Delta \vdash M : \varphi$, the statements in Δ represent hypotheses of the derivation $M^?$ of φ. The mapping '?' will not 'print' these hypotheses in the proof figure it constructs, since it is defined on the structure of the term. To overcome this deficiency, we define an additional mapping that turns the statements of (the rightmost) proof context (of a generalized context) into hypotheses of the proof figure that is to be constructed by '?'.

2.4.2. DEFINITION. **Proof context mapping**

$G' \boxdot \Gamma; p_1 : \psi_1, \ldots, p_i : \psi^i \vdash M : \varphi \Rightarrow$



Applying this proof context mapping before '?' will result in a proof figure of the form prescribed in the definition of derivation with hypotheses; the hypotheses $\psi^1, \ldots, \psi^i$ are declared in the first i lines of the proof.

2.4.3. DEFINITION. **'?', the mapping**

For any proof-term M with $G \vdash M : \varphi$ we define by induction on the structure of M a pre-derivation $M^?$ of φ with respect to G as follows (where $G \equiv \Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n$, $G' \equiv \Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_{n-1}; \Delta_{n-1}$, and $F \equiv \Gamma_1 \boxdot \ldots \boxdot \Gamma_n$).

proof var

$$G \vdash p_i : \psi \quad \Rightarrow \quad \psi^i$$

object var

$$F \vdash x_i : Prop \quad \Rightarrow \quad x \in Prop^i$$

term

$$F \vdash t : Prop \quad \Rightarrow \quad t \in Prop$$

axiom constant

$$G \vdash c : \forall x : Prop.((x \supset \bot) \supset \bot) \supset x \quad \Rightarrow \quad \forall x \in Prop.((x \supset \bot) \supset \bot) \supset x$$

proof-abstraction

$$G \vdash \lambda p_i : \psi.N : \psi \supset \chi \quad \Rightarrow$$

```
| ψ^i
|-----   N^? (on G' ⊡ Γ_n; Δ_n, p_i : ψ)
| ⋮      |
|        |
| χ      |
ψ ⊃ χ
```

object-abstraction

$$G \vdash \lambda x_i : Prop.N : \forall x : Prop.\psi \quad \Rightarrow$$

```
| x(∈ Prop)^i
|------------   N^? (on G' ⊡ Γ_n, x_i : Prop; Δ_n)
| ⋮             |
|               |
| ψ             |
∀x ∈ Prop.ψ
```

proof application

$$G \vdash MN : \psi \quad \Rightarrow$$

```
         N^? (on G)
⋮        |
         |
χ        |
         M^? (on G)
⋮        |
         |
χ ⊃ ψ    |
ψ
```

object application

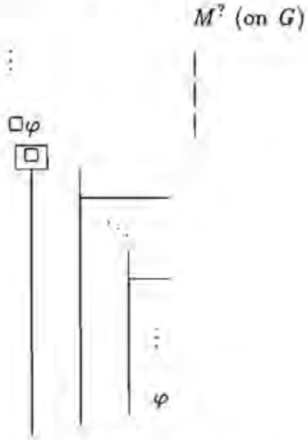
$$G \vdash Nt : \psi \quad \Rightarrow$$

```
                t^? (on F)
t ∈ Prop        |
                N^? (on G)
⋮               |
                |
∀x ∈ Prop.ψ     |
ψ[t/x]
```

K-import

$$G \boxdot \Gamma; \Delta \vdash \check{k}M : \varphi$$

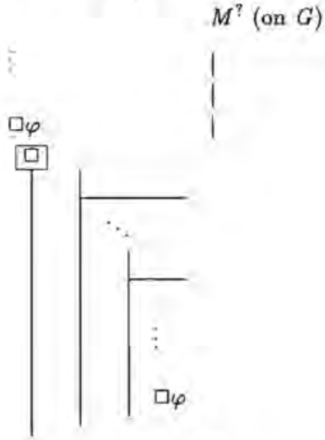
$$\Rightarrow$$



4-import

$$G \boxdot \Gamma; \Delta \vdash \check{4}M : \Box\varphi$$

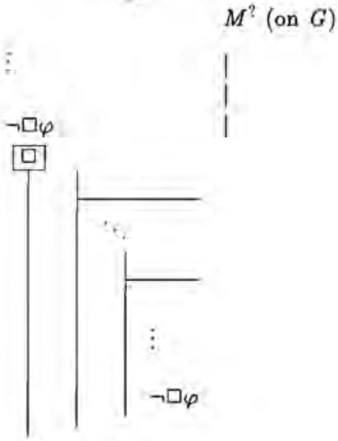
$$\Rightarrow$$



5-import

$$G \boxdot \Gamma; \Delta \vdash \check{5}M : \neg\Box\varphi$$

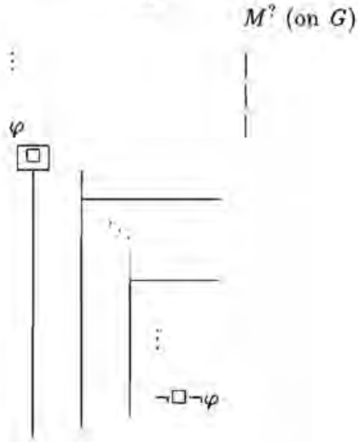
$$\Rightarrow$$



B-import

$$G \boxdot \Gamma; \Delta \vdash \check{b}M : \neg\Box\neg\varphi$$

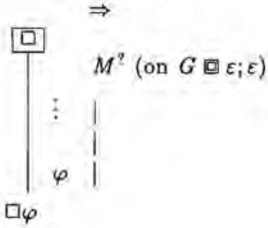
$$\Rightarrow$$



K-export

$$G \vdash \tilde{k}M : \Box\varphi$$

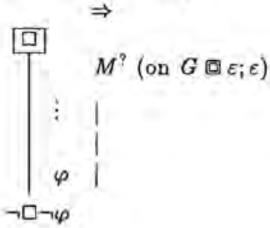
$$\Rightarrow$$



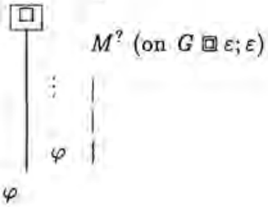
D-export

$$G \vdash \tilde{d}M : \neg\Box\neg\varphi$$

$$\Rightarrow$$

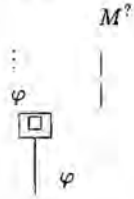


T-export
$G \vdash \hat{t}M : \varphi$
$\Rightarrow$



The mapping of the import terms deserves some explanation, since its effects on the proof figure under construction are not 'local'. The idea is that an import term, say $\check{k}M$, derivable on a generalized context $(G \boxdot \Gamma; \Delta)$ represents a proof of its type, φ, inside a modal interval

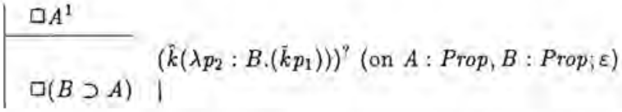
The proof of φ in this position consists in a preceding proof of $\Box\varphi$ in the interval immediately to the left of the modal interval, followed by K-import. Hence '?' decomposes $\check{k}M$ by mapping $M : \Box\varphi$ derivable on G to a proof $(M^?)$ of $\Box\varphi$ just above the modal interval



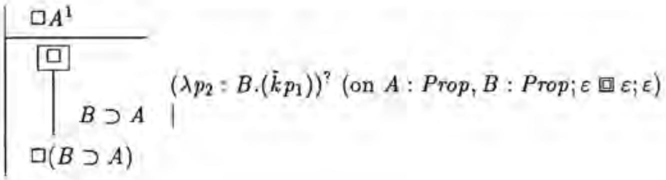
Note that the modal interval is not introduced by mapping the import term, it is supposed to be already present in the proof figure under construction. In an earlier stage of the mapping, an export step is assumed to have introduced the modal interval in which $\check{k}M : \varphi$ is to be evaluated by '?'. To define the mapping of the import terms for the general case that the subordinate context is not empty $(G \boxdot \Gamma; \Delta)$, '?' allows that an unspecified number of hypothesis intervals is already present inside the modal interval.

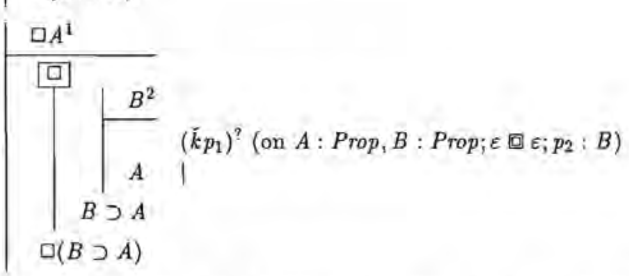
The definition of '?' is justified by the Stripping Lemma, which says that the proposition φ is always equal to a proposition of the form we require. We illustrate the use of the modal rules by means of an example.

2.4.4. EXAMPLE. In this example we construct a proof figure for $A : Prop, B : Prop; p_1 : \Box A \vdash \hat{k}(\lambda p_2 : B.(\check{k}p_1)) : \Box(B \supset A)$. Clearly the proof context Δ is not empty, so before we unravel the term, we have to apply the rule for free proof variables:
$A : Prop, B : Prop; p_1 : \Box A \vdash \hat{k}(\lambda p_2 : B.(\check{k}p_1)) : \Box(B \supset A) \Rightarrow$

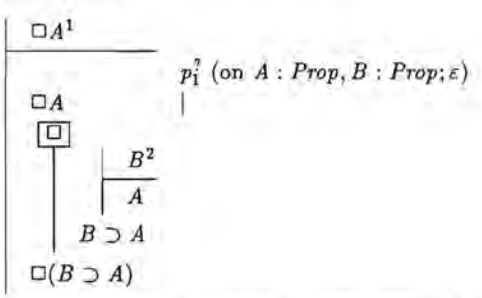


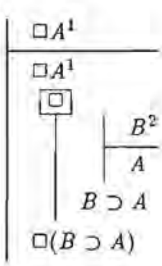
Now that the proof context is empty we can start the mapping of the term by applying the rule for K-export terms, which demands that we open a modal subordinate proof:





After an application of the rule for abstraction that puts the hypothesis B in place, we are faced with a K-import term $\breve{k}p_1$ proving A in a subordinate proof inside a modal proof. '?' decomposes this term by inserting the proof $p_1^?$ of $\Box A$ in the proof figure directly above the modal subordinate proof.





The proofterm for $\Box A$ turns out to be the proof variable $p_1^?$, hence an occurrence of $\Box A^1$ is inserted and we are done.

The reader will have noted that we end up with a natural deduction proof that contains two occurrences of $\Box A^1$ in the same interval, where the topmost occurrence would suffice. The reason for this is that in every step '?' acts on the current structure of the term. Hence the atomic proof object p_1 is mapped to $\Box A^1$, since '?' cannot 'see' that the the hypothesis interval in which $p_1^?$ is to be processed already contains an ocurrence of $\Box A^1$. This 'duplication' will be discussed later when we try to establish an isomorphism between terms and deduction proofs. For the moment we simply remark that these duplications are harmless since they are allowed by the reiteration rule in the natural deduction system.

2.4.5. THEOREM. *If* $\Gamma; \Delta \vdash M : \varphi$ *for a proof term* M *in* $\Lambda\Box PROP2$, *then*

1 $M^?$ *is a deduction proof of* φ *in* $\Box PROP2$ *and all non-discharged assumptions of* $M^?$ *are declared in* Δ,

2 $(M^?)^! \equiv M$.

Both parts of the theorem are basically proved by induction on the structure of M, but since these proofs are not straightforward we discuss them separately.

The problem with the first part of the theorem is that we cannot prove it directly by the desired induction, since unravelling export steps in the term will take us from the simple context $\Gamma; \Delta$ to a generalized context (at least $\Gamma; \Delta \boxdot \varepsilon; \varepsilon$) where we no longer have a induction hypothesis. Hence the theorem must be proved as corollary of something more general.

2.4.6. LEMMA. **Pre-derivation Lemma**
If $G \vdash M : \varphi$ *for a proof term* M *in* $\Lambda\Box PROP2$, *then* $M^?$ *is a pre-derivation of* φ *with respect to* G.

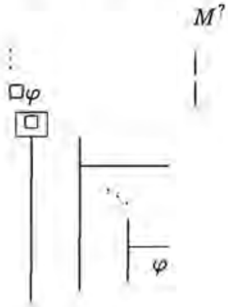
PROOF. By induction on the structure of the term. We show the basic case and two modal cases.

$M \equiv p_i$ is a proof variable.
$G \vdash p_i : \varphi$, applying '?' yields a partial proof figure consisting of one formula:

$$\varphi^i$$

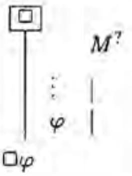
This is only a pre-derivation with respect to G if φ^i can be construed as a reiteration of the hypothesis φ^i, in other words if $p_i : \varphi$ is an element of the rightmost proof context Δ_n. But this is garanteed by Stripping.

$M \equiv \check{k}M$ is an import term.
$G \vdash \check{k}M : \varphi$. By Stripping, $G' \vdash M : \Box\varphi$ where $G \equiv G' \boxdot \Gamma; \Delta$, hence by IH $M^?$ is a pre-derivation of $\Box\varphi$ with respect to G'. Since the context G contains a $\boxdot$ immediately following G' $M^?$ is a pre-derivation of $\Box\varphi$ in an interval immediately to the left of a modal interval. Hence K-import can be applied to $\Box\varphi$. This rule allows import of a formula over an arbitrary number of hypothesis-intervals inside the modal interval



Therefore we can apply K-import to obtain a pre-derivation $(\check{k}M)^?$ of φ with respect to $G' \boxdot \Gamma; \Delta (\equiv G)$.

$M \equiv \hat{k}M$ is an export term.
$G \vdash \hat{k}M : \Box\varphi$. By Stripping $G \boxdot \varepsilon; \varepsilon \vdash M : \varphi$, hence by IH $M^?$ is a pre-derivation of φ with respect to $G \boxdot \varepsilon; \varepsilon$. Since the context contains a $\boxdot$, $M^?$ is a pre-derivation of φ inside a modal interval. Moreover, the subordinate context is empty $(\varepsilon; \varepsilon)$ and so φ lies in the modal interval (there are no undischarged hypotheses):



Therefore we can apply K-export to obtain a pre-derivation $(\hat{k}M)^?$ of $\Box\varphi$ with respect to G.

Given the pre-derivation lemma and the following fact, we can prove that '?' yields deduction proofs in $\Box PROP2$.

2.4.7. FACT. A pre-derivation of which the conclusion, C, is not an element of a modal interval $(degree(C) = (0, i)$ for some $i)$ is a derivation (with hypotheses) in $\Box PROP2$.

2.4.8. THEOREM. (part 1)
If $\Gamma; \Delta \vdash M : \varphi$ *for a proof term* M *in* $\lambda\Box PROP2$, $M^?$ *is a derivation of* φ *in* $\Box PROP2$, *and all hypotheses of* $M^?$ *are declared in* Δ.

PROOF. By the pre-derivation lemma, we have that $M^?$ is a pre-derivation of φ with respect to $\Gamma; \Delta$. Since $\Gamma; \Delta$ is a simple context, the conclusion of $M^?$ is not an element of a modal interval. Hence by the fact above, $M^?$ is a derivation (possibly with hypotheses) of φ. If $M^?$ has hypotheses, these correspond to proof variables occurring freely in M. By the meta theory of $\lambda\Box PROP2$ these variables are all declared in Δ[1], and hence all hypotheses of $M^?$ are declared in Δ.

To prove the second part of the theorem, we have to take an additional feature of natural deductions into account. Fitch deduction proofs can contain 'blind alleys', the rules allow the occurrence of (series of) deduction steps that do not contribute in any way to proving the goal formula. A simple example of this is that the reiteration rules allows you to repeat assumptions anywhere in their hypothesis interval, regardless whether this reiterated formula will be used in the remainder of the proof. Sequences of useless steps can also appear, for instance as a result of the duplication procedure described in the formulation of OK-deductions. The following is an example of a modal proof with a blind alley:

[1]Cf. the free variable lemma for MPTSs in chapter 3.

1.	$\Box(\chi \supset \varphi)$	
2.	$\Box\chi$	
3.	$\Box(\chi \supset \varphi)$	(reiteration 1)
	$\boxed{\Box}$	
4.	$\chi \supset \psi$	
5.	$\chi \supset \varphi$	(K-import 3)
6.	χ	(Import 2)
7.	φ	($\supset$-elim 5,6)
8.	ψ	($\supset$-elim 4,6)
9.	$(\chi \supset \psi) \supset \psi$	($\supset$-intro 4-8)
10.	$\Box((\chi \supset \psi) \supset \psi)$	(K-export 9)
11.	$\Box\chi \supset \Box((\chi \supset \psi) \supset \psi)$	($\supset$-intro 2-10)
12.	$\Box(\chi \supset \varphi) \supset (\Box\chi \supset \Box((\chi \supset \psi) \supset \psi))$	($\supset$-intro 1-11)

This proof contains a useless derivation leading from the assumptions to φ at line 7. The mapping '!' ignores this blind alley, which can be seen from looking at the resulting term $\lambda p_1 : \Box(\chi \supset \psi).\lambda p_2 : \Box\chi.\hat{k}(\lambda p_3 : \chi \supset \psi.(p_3 \check{k} p_2))$. It does not contain a subterm of the form $\check{k}p_2\check{k}p_1$, corresponding to the application of (the K-imported hypotheses) $\chi \supset \varphi$ and χ in line 7 of the proof. Mapping this term back to a natural deduction proof leads to the following proof figure from which the entire blind alley (lines 3,5, and 7 above) is missing.

1.	$\Box(\chi \supset \varphi)^1$
2.	$\Box\chi^2$
	$\boxed{\Box}$
3.	$\chi \supset \psi^3$
4.	$\chi \supset \psi^3$
5.	χ
6.	ψ
7.	$(\chi \supset \psi) \supset \psi$
8.	$\Box((\chi \supset \psi) \supset \psi)$
9.	$\Box\chi \supset \Box((\chi \supset \psi) \supset \psi)$
10.	$\Box(\chi \supset \varphi) \supset (\Box\chi \supset \Box(\chi \supset \psi) \supset \psi))$

The reason that '!' removes blind alleys, is that it starts from the conclusion of the deduction proof and inductively works its way upward. Hence formulas that are not involved in applications of deduction rules in a backward path from the conclusion are ignored. Given the notion of blind alley and the knowledge that they are ignored by '!', we can prove that $(M^!)^? \equiv M$.

2.4.9. THEOREM. (part 2)
If $\Gamma; \Delta \vdash M : \varphi$ *for a proof term* M *in* $\Lambda\Box PROP2$, *then* $(M^!)^? \equiv M$.

PROOF. By induction on the structure of the term. Some reasoning about the proof intervals over which the $M^?$ 'range' is required in the cases for application, abstraction and free proof variables. We do an application case, the others are similar.

$M \equiv M_1 M_2$ is an application (of proof terms).

$G \vdash M_1 M_2 : \psi \Rightarrow^?$

In the proof figure on the left resulting from applying '?', the proofs $M_1^?$ and $M_2^?$ are disjoint. However, for '!' the situation is as depicted in the figure on the right, where $M_2^?.M_1^?$ proving $\varphi \supset \psi$ is $M_2^?$ followed by $M_1^?$. Hence applying '!' leaves us with (1) $G_1 \vdash (M_2^?.M_1^?)^! : \varphi \supset \psi$ and (2) $G_2 \vdash (M_2^?)^! : \varphi$. However, by the fact that in the proof $M_1^?$ of $\varphi \supset \psi$ no rules are applied to formulas of $M_2^?$, we can conclude that $M_2^?$ is a blind alley in $M_2^?.M_1^?$. Therefore '!' will remove it and $(M_2^?.M_1^?)^! \equiv (M_1^?)^!$. The rest is straightforward:

(1) $G_1 \vdash (M_1^?)^! : \varphi \supset \psi$ by IH $(M_1^?)^! \equiv M_1$, hence $G_1 \vdash M_1 : \varphi \supset \psi$

(2) $G_2 \vdash (M_2^?)^! : \varphi$ by IH $(M_2^?)^! \equiv M_2$, hence $G_2 \vdash M_2 : \varphi$

Therefore, $G_1 \cup_{zip} G_2 \vdash M_1 M_2 : \psi$.

If we want to prove a 'back and forth'-equivalence for mapping natural deductions to terms to natural deductions, we will have to deal with a number of ways in which the composition of '!' and '?' transforms natural deduction proofs. These transformations indicate how the class of 'OK-deductions' has to be restricted to obtain an invariance result.

We have just seen such a transformation, the combination of ! and ? 'cleans' natural deductions proofs by removing blind alleys. This indicates that a back and forth equivalence can only be reached on a class of 'clean proofs'. Using the annotation of the natural deduction proofs, a definition of 'clean proof' and a method for 'cleaning' proofs can be given. The idea is that since the annotation of the proof figures records the applied rule as well as the line numbers of the premisses of that rule, we can tell that a formula appearing in a proof is not used in the rest of that proof if its line number does not appear in the annotation of any of the lines of the proof.

2.4.10. DEFINITION. **Set of used line numbers**
Given a proof figure **D**, the *set of used line numbers* in **D** is:
$\mathcal{N}^D = \{i | i$ appears in the annotation of $\mathbf{D}\ \}$

2.4.11. DEFINITION. **Clean proofs**
A proof figure **D** consisting of a closed interval $D = [1, n]$ ($D \subset IN$) is *clean* when for all line numbers j $(1 \leq j < n) : (j = k$ for some $[k, l] \in\mathbf{H}$ or $j \in \mathcal{N}^D)$.

Note that $j < n$ since the last line always contains the conclusion, and that the definition allows for assumptions (topmost formulas of hypothesis intervals) that are not used in the proof. 'Cleaning' a proof simply consists in removing the unused formulas.

2.4.12. DEFINITION. **Cleaning** ($\Rightarrow^{clean}$)
Given a proof figure **D** consisting of a closed interval $D = [1, n]$ ($D \subset IN$), the *cleaning operation* $\Rightarrow^{clean}$ removes all lines j $(1 \leq j < n)$ for which ($j \notin \mathcal{N}^D$ and line j does not contain a hypothesis).

Cleaning will remove blind alleys consisting of a series of deduction steps when applied repeatedly: after the removal of the 'conclusion' of the blind alley, the line numbers of the premisses used in the last rule application will disappear from the set $\mathcal{N}^D$ since they only appeared in the annotation of the line of the conclusion. We will demonstrate this using the modal proof with blind alley discussed above. In order to avoid complicated renumbering operations on the annotation in between applications of cleaning, we will compare the set $\mathcal{N}^D$ with the set $\mathcal{N}^l$ of *line numbers of the proof figure*, instead of the interval D.

The proof figure contains a useless derivation of φ at line 7. Comparing $\mathcal{N}^D = \{1, 2, 3, 4, 5, 6, 8, 9, 10, 11\}$ with $\mathcal{N}^l = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$ shows that, except for 12 (the last line), the only 'missing number' is 7. Applying $\Rightarrow^{clean}$ removes this line:

1.	$\Box(\chi \supset \varphi)$	
2.	$\Box\chi$	
3.	$\Box(\chi \supset \varphi)$	(reiteration 1)
	$\boxed{\Box}$	
4.	$\chi \supset \psi$	
5.	$\chi \supset \varphi$	(Import 3)
6.	χ	(Import 2)
8.	ψ	($\supset$-elim 4,6)
9.	$(\chi \supset \psi) \supset \psi$	($\supset$-intro 4-8)
10.	$\Box((\chi \supset \psi) \supset \psi)$	(K-export 9)
11.	$\Box\chi \supset \Box((\chi \supset \psi) \supset \psi)$	($\supset$-intro 2-10)
12.	$\Box(\chi \supset \varphi) \supset (\Box\chi \supset \Box((\chi \supset \psi) \supset \psi))$	($\supset$-intro 1-11)

Since removing line 7 also removes its annotation, we can now see that the Import of $\chi \supset \varphi$ at line 5 is pointless: $\mathcal{N}^D = \{1, 2, 3, 4, 6, 8, 9, 10, 11\}$ and $\mathcal{N}^l = \{1, 2, 3, 4, 5, 6, 8, 9, 10, 11, 12\}$.

$\Rightarrow^{clean}$

1.	$\Box(\chi \supset \varphi)$	
2.	$\Box\chi$	
3.	$\Box(\chi \supset \varphi)$	(reiteration 1)
	$\boxed{\Box}$	
4.	$\chi \supset \psi$	
6.	χ	(Import 2)
8.	ψ	($\supset$-elim 4,6)
9.	$(\chi \supset \psi) \supset \psi$	($\supset$-intro 4-8)
10.	$\Box((\chi \supset \psi) \supset \psi)$	(K-export 9)
11.	$\Box\chi \supset \Box((\chi \supset \psi) \supset \psi)$	($\supset$-intro 2-10)
12.	$\Box(\chi \supset \varphi) \supset (\Box\chi \supset \Box((\chi \supset \psi) \supset \psi))$	($\supset$-intro 1-11)

The set $\mathcal{N}^D = \{1, 2, 4, 6, 7, 8, 9\}$ of the resulting proof figure misses 3. This is correct since removing the imported occurence of $\chi \supset \varphi$ makes the reiteration of the assumption $\Box(\chi \supset \varphi)$ superfluous.

$$\Rightarrow^{clean}$$

1.	$\Box(\chi \supset \varphi)$	
2.	$\Box\chi$	
	$\boxed{\Box}$	
4.	$\chi \supset \psi$	
6.	χ	(Import 2)
8.	ψ	($\supset$-elim 4,6)
9.	$(\chi \supset \psi) \supset \psi$	($\supset$-intro 4-8)
10.	$\Box((\chi \supset \psi) \supset \psi)$	(K-export 9)
11.	$\Box\chi \supset \Box((\chi \supset \psi) \supset \psi)$	($\supset$-intro 2-10)
12.	$\Box(\chi \supset \varphi) \supset (\Box\chi \supset \Box((\chi \supset \psi) \supset \psi))$	($\supset$-intro 1-11)

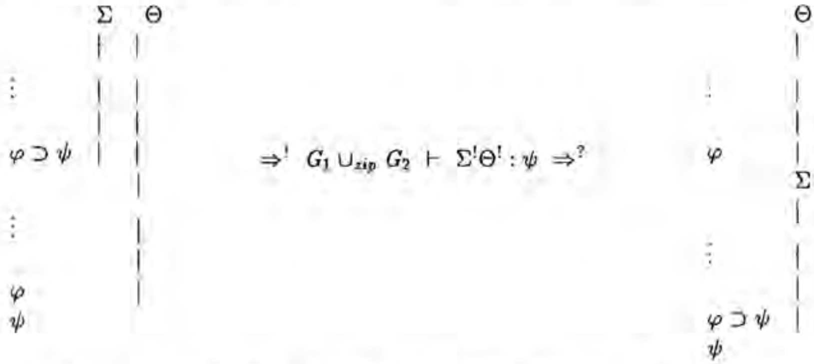
The blind alley from the original deduction has now completely been removed, and the cleaning operation halts since $\mathcal{N}^D = \{1, 2, 4, 6, 8, 9, 10, 11\}$ and $\mathcal{N}^I = \{1, 2, 4, 6, 8, 9, 10, 11, 12\}$ (the only missing number is that of the last line, 12).

Cleaning up is still not sufficient to get a managable class of deductions which are invariant under the composed mappings. There is a degree of freedom in the definition of the application of the elimination rules for $\supset$ and $\forall$, that can cause the deduction resulting from the composed mappings to differ substantially from the original:

> $R \in \{\supset\text{-}elim, \forall\text{-}elim\}$.
> In this case the premisses and the conclusion E all lie in the same interval.
> *The order in which the premisses appear is free.*

Under the type theoretical rules the order of the subterms proving $\varphi \supset \psi$ and φ (or $\forall x : Prop.\psi$

and $t \in Prop$) cannot be varied, the 'function term' of type $\varphi \supset \psi$ must be applied to the 'argument term' of type φ. This means that under the combined mappings parts of the proof figure may get interchanged:



However, we can drop the clause allowing free order of arguments for $\supset$-*elim* and $\forall$-*elim*. It is obvious that if Σ proves φ with free order of arguments for these rules we can find a Σ' which does the same with arguments in fixed order.

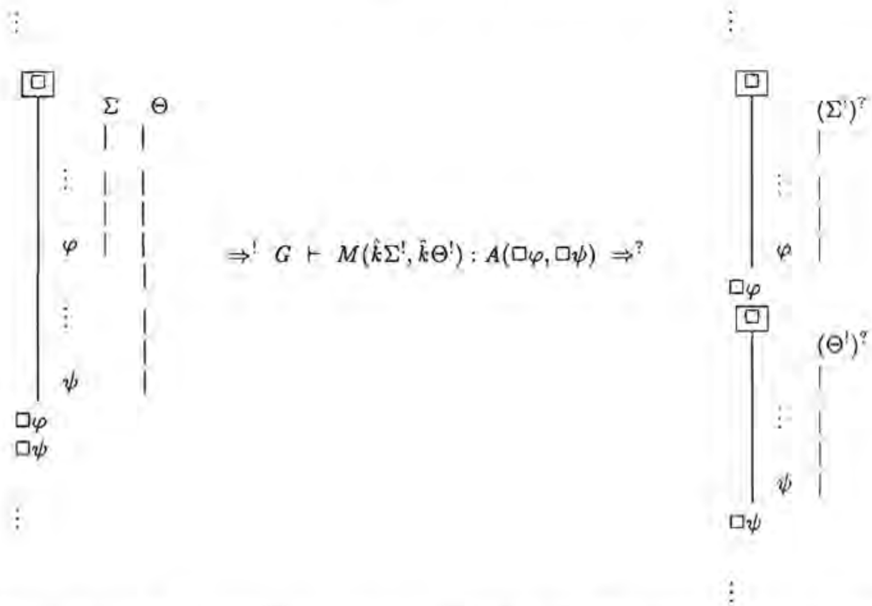
$R = \{\supset\text{-}elim)\}$.
In this case the premisses $P_1 = \varphi$, $P_2 = \varphi \supset \psi$ and the conclusion $E = \psi$ all lie in the same interval, and P_1 precedes P_2.

$R = \{\forall\text{-}elim)\}$.
In this case the premisses $P_1 = t \in Prop$, $P_2 = \forall x \in Prop.\psi$ and the conclusion $E = \psi$ all lie in the same interval, and P_1 precedes P_2.

Since the order 'argument' above 'function' is now fixed in the deduction proof and the ? mapping puts the subterms representing them back in the same order, these parts of the proof figure can no longer get interchanged.

The modal rules can also cause trouble under the combined mappings. The rules of $\Box PROP2$ allow 'multiple export': it is possible to export more than one formula from a modal subordinate proof. Unfortunately, this practice is not supported by the combined mappings.

A schematical example of multiple export is the following deduction (where $\check{k}(\Sigma^!)$ and $\check{k}(\Theta^!)$ appear in M and $\Box\varphi$ and $\Box\psi$ may appear in A):



Since '!' maps proof figures to terms 'from the conclusion upward', applying it to a multiple export results in a term in which several export subterms appear: ($\breve{k}\Sigma$ and $\breve{k}\Theta^{!}$). The '?'-mapping will then open a new modal subordinate modal proof from each of these subterms in the (re)construction of the proof figure instead of reuniting $(\Sigma^{!})^{?}$ and $(\Theta^{!})^{?}$ in one subordinate modal proof. However, from the example we can already see that nothing is lost by restricting export to only one formula per modal subordinate proof. At worst we have to duplicate the original modal subordinate proof for each of the formulas originally part of the multiple export.

Combining the above observations we arrive at a restricted class of OK proofs that seems suitable for proving a 'back-and-forth' equivalence.

2.4.13. DEFINITION. **A-OK proofs**
A natural deduction proof Σ of φ is *A-OK* iff:

Σ is OK

Σ is clean

Σ has ordered premisses for $\forall$-*elim* and $\supset$-*elim*

Σ has no multiple exports

Before we can establish a bijection between terms and proof figures we have to take care of one last transformation of the natural deduction proofs caused by the composition of the mappings. As noticed in earlier examples, the combined mappings can lead to 'duplication' of formulas. The following simple example shows this effect:

$$
\begin{array}{|l}
\varphi \supset \psi^1 \\ \hline
\begin{array}{|l}
\varphi^2 \\ \hline
\varphi \supset \psi^1 \\
\psi
\end{array} \\
\varphi \supset \psi
\end{array}
$$
$$(\varphi \supset \psi) \supset (\varphi \supset \psi)$$

$$\Rightarrow^{!} \ \Gamma_\varphi \cup \Gamma_{\varphi \supset \psi}; \varepsilon \ \vdash \ \lambda p_1 : \varphi \supset \psi.(\lambda p_2 : \varphi.(p_1 p_2)) : (\varphi \supset \psi) \supset (\varphi \supset \psi) \ \Rightarrow^{?}$$

$$
\begin{array}{|l}
\varphi \supset \psi^1 \\ \hline
\begin{array}{|l}
\varphi^2 \\ \hline
\varphi^2 \\
\varphi \supset \psi^1 \\
\psi
\end{array} \\
\varphi \supset \psi
\end{array}
$$
$$(\varphi \supset \psi) \supset (\varphi \supset \psi)$$

After application of '!'and '?' we end up with an additional occurrence of φ^2. The inductive mapping '!' generates two occurrences of p_2 in the term: one corresponding to the occurrence of φ^2 as the hypothesis of the inner subordinate proof, and one corresponding to the use of φ^2 as a premisse for $\supset$-elimination. The mapping '?' acts on the structure of the term, first the abstraction over p_2 is mapped to a subordinate proof with hypothesis φ^2, later the (sub)term consisting of the proof variable p_2 is mapped to reiteration of φ^2 in that hypothesis interval. Since '?' does not take the partial proof figure into account that it has already constructed, it is too 'shortsighted' to see that it is duplicating φ^2.

Unlike the previous cases, this transformation cannot be excluded by restricting the class of proofs on which the mappings are to be applied. However we can deal with this relatively innocent phenomenon by introducing an equivalence relation on A-OK proofs: two natural deduction proofs are equivalent if they are identical when stripped of doubles.

2.4.14. DEFINITION. **Double occurrence**
A formula φ *occurs doubly* in a proof figure $\mathbf{D}$ iff $\exists I \in (\mathbf{H} \cup D)$ such that $\exists i, j \in I(F(i) = F(j))$.

2.4.15. DEFINITION. **Doubles normal form**
An A-OK natural deduction proof Σ of φ is in *doubles normal form* iff no formula ψ occurs doubly in it.

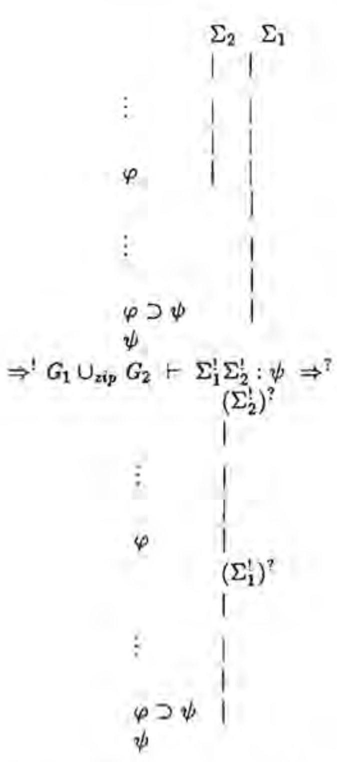
2.4.16. DEFINITION. **Doubles equivalence**
Two A-OK natural deduction proofs Σ and Θ of a formula φ are *doubles equivalent*, $\Sigma =_{(doubles)} \Theta$, iff they have the same doubles normal form.

A proof figure can easily be brought in doubles normal form by simply deleting for each doubly occurring formula (say $F(i) = F(j)$ and $i < j$) the occurrence at the line with the higher number (j).

2.4.17. PROPOSITION. *Let Σ be an A-OK natural deduction proof in $\Box PROP2$, then*

$$(\Sigma^{!})^{?} =_{doubles} \Sigma$$

PROOF. By induction on the derivation Σ. Doubles equivalence is used in the cases for abstraction, application and proof assumption. We show the case for $\supset$-elim.



By IH $(\Sigma_1^{!})^{?} =_{doubles} \Sigma_1$ and $(\Sigma_2^{!})^{?} =_{doubles} \Sigma_2$. Hence the only difference between Σ and $(\Sigma^{!})^{?}$ is that in Σ the subderivations Σ_1 and Σ_2 can overlap, allowing formulas of Σ_1 to be used in Σ_2. Since by IH $(\Sigma_1^{!})^{?} =_{doubles} \Sigma_1$ and $(\Sigma_2^{!})^{?} =_{doubles} \Sigma_2$, we have that in $(\Sigma^{!})^{?}$ some occurrences of formulas in $(\Sigma_1^{!})^{?}$ may be doubled in $(\Sigma_2^{!})^{?}$. But then $(\Sigma^{!})^{?} =_{doubles} \Sigma$.

2.4.18. COROLLARY. *The mappings '!' and '?' constitute a bijection between $=_{doubles}$-equivalence classes of A-OK deductions in $\Box PROP2$ and pairs (Δ, M) in $\Lambda\Box PROP2$.*

PROOF. By the result above and the earlier back-and-forth theorem for terms of $\Lambda\Box PROP2$

2.5. From $\Lambda\Box PROP2$ to $\lambda\Box PROP2$ and back

The last step in mapping modal natural deduction to modal pure type sytems is to show that the system $\Lambda\Box PROP2$ is equivalent to the MPTS $\lambda\Box PROP2$. In [Geuvers 1993] the proof of equivalence of the intermediate and the 'target' type system of the interpretation hinges on the following basic property of PTSs in the Logic Cube.

2.5.1. PROPOSITION. *In $\lambda PRED\omega$ we have the following.*
If $\Gamma \vdash M : A$ then $\Gamma_D, \Gamma_T, \Gamma_P \vdash M : A$ where

- *$\Gamma_D, \Gamma_T, \Gamma_P$ is a sound permutation of Γ (i. e. , it is a legal context that is a permutation of Γ),*
- *Γ_D only contains declarations of the form $x : Set$,*
- *Γ_T only contains declarations of the form $x : A$ with $\Gamma_D \vdash A : Set/Type^p$,*
- *Γ_P only contains (ordered) declarations of the form $x : \varphi$ with $\Gamma_D, \Gamma_T \vdash \varphi : Prop$,*
- *if $A \equiv Set/Type^p$, then $\Gamma_D \vdash M : A$,*
- *if $\Gamma \vdash A : Set/Type^p$, then $\Gamma_D, \Gamma_T \vdash M : A$.*

PROOF. By induction on the derivation

For the subsystem $\lambda PROP2$, this proposition allows us to split any context Γ into *two* parts. In $\lambda PROP2$ *Set* is not a sort, hence Γ_D is always empty.

2.5.2. PROPOSITION. *In $\lambda PROP2$ we have the following.*
If $\Gamma \vdash M : A$ then $\Gamma_T, \Gamma_P \vdash M : A$ where

- *Γ_T, Γ_P is a sound permutation of Γ*
- *Γ_T only contains declarations of the form $x : Prop$,*
- *Γ_P only contains (ordered) declarations of the form $x : \varphi$ with $\Gamma_T \vdash \varphi : Prop$,*
- *if $A \equiv Type^p$, then $\varepsilon \vdash M : A$ with $M \equiv Prop$,*
- *if $\Gamma \vdash A : Type^p$, then $\Gamma_T \vdash M : A$ with $A \equiv Prop$.*

In the generalized contexts of the MPTS $\lambda\Box PROP2$ we would like to have these properties for each of the constituting contexts $\Gamma_1, \dots \Gamma_n$:

2.5.3. PROPOSITION. *In $\lambda\Box PROP2$ we have the following.*
If $G \vdash M : A$, $G \equiv \Gamma_1 \boxdot \dots \boxdot \Gamma_n$, then $\Gamma^1_T, \Gamma^1_P \boxdot \dots \boxdot \Gamma^n_T, \Gamma^n_P \vdash M : A$ where

- *Γ^i_T, Γ^i_P is a permutation of Γ^i for all $i : 1 \leq i \leq n$ and $\Gamma^1_T, \Gamma^1_P \boxdot \dots \boxdot \Gamma^n_T, \Gamma^n_P$ is legal,*
- *Γ^i_T only contains declarations of the form $x : Prop$ for all $i : 1 \leq i \leq n$, ,*
- *Γ^i_P only contains (ordered) declarations of the form $x : \varphi$ with $\Gamma^1_T \boxdot \dots \boxdot \Gamma^i_T \vdash \varphi : Prop$,*

- *if* $A \equiv Type^p$, *then* $\varepsilon \vdash M : A$ *with* $A \equiv Prop$,
- *if* $\Gamma_1 \boxdot \ldots \boxdot \Gamma_n \vdash A : Type^p$, *then* $\Gamma_T^1 \boxdot \ldots \boxdot \Gamma_T^n \vdash M : A$ *with* $A \equiv Prop$.

PROOF. By induction on the derivation (using the Strong Permutation Lemma[2] for $MPTS_\beta$ in the subcase of Weakening where $s \equiv Type^p$).

The lemma clearly shows the similarities between $\Lambda\Box PROP2$ and $\lambda\Box PROP2$, just read 'Γ_T^i' for 'Γ^i' and 'Γ_P^i' for 'Δ^i'. Hence we would like to prove:

1 $\Gamma_1 \boxdot \ldots \boxdot \Gamma_n \vdash M : Prop$ (*in* $\Lambda\Box PROP2$) $\Leftrightarrow$
$\Gamma_T^1 \boxdot \ldots \boxdot \Gamma_T^n \vdash M : Prop$ (*in* $\lambda\Box PROP2$)

2 $\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n \vdash M : A(: Prop)$ (*in* $\Lambda\Box PROP2$) $\Leftrightarrow$
$\Gamma_T^1, \Gamma_P^1 \boxdot \ldots \boxdot \Gamma_T^n, \Gamma_P^n \vdash M : A(: Prop)$ (*in* $\lambda\Box PROP2$)

However, the syntax of $\Lambda\Box PROP2$ differs slightly from that of $\lambda\Box PROP2$ so we first define translations mappisg pseudo terms of either system to pseudo terms of the other system.

2.5.4. DEFINITION. **Translation $\Lambda\Box PROP2 \Rightarrow \lambda\Box PROP2$**
Pseudoterms and generalized contexts of $\Lambda\Box PROP2$ are mapped to pseudoterms and generalized contexts of $\lambda\Box PROP2$ by the inductively defined mapping '$\sharp$', where the variables of $\lambda\Box PROP2$ are partitioned into 'kind variables' ($x^{Type^p} : Prop$) and 'type variables' ($x^{Prop} : \varphi : Prop$).

Object terms

$$Prop^\sharp = Prop$$
$$(x_i)^\sharp = x_i^{Type^p} \quad (x_i \in Var^{ob})$$
$$(T \supset T)^\sharp = \Pi x^{Prop} : T^\sharp . T^\sharp$$
$$(\forall x : Prop.T)^\sharp = \Pi x^\sharp : Prop.T^\sharp$$
$$(\Box T)^\sharp = \Box T^\sharp$$

Proof terms

$$(p_i)^\sharp = x_i^{Prop} \quad (p_i \in Var^{pr})$$
$$(PrPr)^\sharp = Pr^\sharp Pr^\sharp$$
$$(PrT)^\sharp = Pr^\sharp T^\sharp$$
$$(\lambda p : T.T)^\sharp = \lambda p^\sharp : T^\sharp . T^\sharp$$
$$(\lambda x : Prop.T)^\sharp = \lambda x^\sharp : Prop.T^\sharp$$
$$(c)^\sharp (c \in Constants_{\Lambda\Box PROP2}) = c \ (c \in Constants_{\lambda\Box PROP2})$$
$$(\check{k} T)^\sharp = \check{k} T^\sharp, (\check{4} T)^\sharp = \check{4} T^\sharp, (\check{5} T)^\sharp = \check{5} T^\sharp, (\check{b} T)^\sharp = \check{b} T^\sharp$$
$$(\hat{k} T)^\sharp = \hat{k} T^\sharp, (\hat{d} T)^\sharp = \hat{d} T^\sharp, (\hat{t} T)^\sharp = \hat{t} T^\sharp$$

Contexts

[2] For $G_1, x : A, y : B, \Gamma_2$ a generalized context, M and C terms, with $x \notin FV(B)$, $G_1, x : A, y : B, \Gamma_2 \vdash M : C \Rightarrow G_1, y : B, x : A\Gamma_2 \vdash M : C$. Proof cf. Chapter 3

$$\varepsilon^\sharp = \varepsilon$$
$$(\Gamma, x_i : Prop)^\sharp = \Gamma^\sharp, x_i^\sharp : Prop$$
$$(F \boxdot \Gamma)^\sharp = F^\sharp \boxdot \Gamma^\sharp$$
$$(\Delta, p_i : \varphi)^\sharp = \Delta^\sharp, p_i^\sharp : \varphi^\sharp$$
$$(\Gamma; \Delta)^\sharp = \Gamma^\sharp, \Delta^\sharp$$
$$(G \boxdot \Gamma; \Delta)^\sharp = G^\sharp \boxdot (\Gamma; \Delta)^\sharp.$$

2.5.5. DEFINITION. **Translation $\lambda\Box PROP2 \Rightarrow \Lambda\Box PROP2$**
Pseudoterms and generalized contexs of $\lambda\Box PROP2$ are mapped to pseudoterms and generalized contexts of $\Lambda\Box PROP2$ by the inductively defined mapping '$\flat$', where the variables of $\lambda\Box PROP2$ are partioned into 'kind variables' ($x^{Type^P} : Prop$) and 'type variables' ($x^{Prop} : A : Prop$).

Object terms

$$Prop^\flat = Prop$$
$$(x_i^{Type^P})^\flat = x_i (\in Var^{ob})$$
$$(\Pi x^{Prop} : T.T)^\flat = T^\flat \supset T^\flat$$
$$(\Pi x^{Type^P} : Prop.T)^\flat = \forall (x^{Type^P})^\flat : Prop.T^\flat$$
$$(\Box T)^\flat = \Box T^\flat$$

Proof terms

$$(x_i^{Prop})^\flat = p_i (\in Var^{pr})$$
$$(TT)^\flat = T^\flat T^\flat$$
$$(\lambda x^{Prop} : T.T)^\flat = \lambda (x^{Prop})^\flat : T^\flat.T^\flat$$
$$(\lambda x^{Type^P} : Prop.T)^\flat = \lambda (x^{Type^P})^\flat : Prop.T^\flat$$
$$(c)^\flat (c \in Constants_{\lambda\Box PROP2}) = c \ (c \in Constants_{\Lambda\Box PROP2})$$
$$(\check{k} T)^\flat = \check{k} T^\flat, (\check{4} T)^\flat = \check{4} T^\flat, (\check{5} T)^\flat = \check{5} T^\flat, (\check{b} T)^\flat = \check{b} T^\flat$$
$$(\hat{k} T)^\flat = \hat{k} T^\flat, (\hat{d} T)^\flat = \hat{d} T^\flat, (\hat{t} T)^\flat = \hat{t} T^\flat$$

Contexts

$$\varepsilon^\flat = \varepsilon$$
$$(\Gamma_T, x_i : Prop)^\flat = \Gamma_T^\flat, x_i^\flat : Prop$$
$$(\Gamma_P, x_i : \varphi)^\flat = \Gamma_P^\flat, p_i^\flat : \varphi^\flat$$
$$(\Gamma_T, \Gamma_P)^\flat = \Gamma_T^\flat; \Gamma_P^\flat$$
$$(G \boxdot \Gamma_P, \Gamma_T)^\flat = G^\flat \boxdot (\Gamma_T, \Gamma_P)^\flat.$$

Since both '$\sharp$' and '$\flat$' preserve the indices of the variables we have 'enharmonic equality' for their compositions:

If M is a pseudo term in $\lambda\Box PROP2$, then $(M^\sharp)^\flat \equiv M$
If G is a generalized context in $\lambda\Box PROP2$, then $(G^\sharp)^\flat \equiv G$
If M is a pseudo term in $\Lambda\Box PROP2$, then $(M^\flat)^\sharp \equiv M$
If G is a generalized context in $\Lambda\Box PROP2$, then $(G^\flat)^\sharp \equiv G$

In proving the equivalence of the two systems, we need that the mappings are transparent with respect to substitution.

2.5.6. LEMMA. **Substitution Preservation Lemma**

> $\sharp$ *preserves substitutions:*
> *for* $\Lambda\Box PROP2$*-terms* A *and* B $(A[B/x])^\sharp = A^\sharp[B^\sharp/x^\sharp]$.
>
> $\flat$ *preserves substitutions:*
> *for* $\lambda\Box PROP2$*-terms* A *and* B $(A[B/x])^\flat = A^\flat[B^\flat/x^\flat]$.

PROOF. Both are proved by induction on the structure of the term A.

Now we are ready to prove that both mappings preserve typeability, for any term that is typeable in $\Lambda\Box PROP2$ (with a certain type on a certain context), its translation is typeable in $\lambda\Box PROP2$ (with the translation of the type on the translation of the context) and vice versa. In some cases additional properties of MPTS's are needed. For full formal definitions and proofs of these properties the reader is referred to the next chapter.

2.5.7. THEOREM. *For all terms* M *in* $\Lambda\Box PROP2$*:*

1 *If* $\Gamma_1 \boxdot \ldots \boxdot \Gamma_n \vdash M : Prop$, *then* $(\Gamma_1 \boxdot \ldots \boxdot \Gamma_n)^\sharp \vdash M^\sharp : Prop$

2 *If* $\Gamma_1;\Delta_1 \boxdot \ldots \boxdot \Gamma_n;\Delta_n \vdash M : \varphi(: Prop)$, *then* $(\Gamma_1;\Delta_1 \boxdot \ldots \boxdot \Gamma_n;\Delta_n)^\sharp \vdash M^\sharp : \varphi^\sharp$

PROOF. By induction on the structure of the term (M).

In the proof of 1 the additional property Strong Thinning is needed:

$M \equiv x_i (\in Var^{ob})$
$\Gamma_1 \boxdot \ldots \boxdot \Gamma_n \vdash x_i : Prop$, hence by Stripping $(x_i : Prop) \in \Gamma_i$ for some $i:\ 1 \leq i \leq n$, but then (by definition) $(x_i)^\sharp \in (\Gamma_i)^\sharp$ and so by the Start lemma:
$(\Gamma_1)^\sharp \boxdot \ldots \boxdot (\Gamma_i)^\sharp \vdash (x_i)^\sharp : Prop$ and hence by Strong Thinning:
$(\Gamma_1)^\sharp \boxdot \ldots \boxdot (\Gamma_i)^\sharp \boxdot \ldots \boxdot (\Gamma_n)^\sharp \vdash (x_i)^\sharp : Prop$. Since by definition
$(\Gamma_1)^\sharp \boxdot \ldots \boxdot (\Gamma_n)^\sharp = (\Gamma_1 \boxdot \ldots \boxdot \Gamma_n)^\sharp$ we have
$(\Gamma_1 \boxdot \ldots \boxdot \Gamma_n)^\sharp \vdash (x_i)^\sharp : Prop$.

In the proof of 2, the case of abstraction over a propositional variable causes trouble. In $\Lambda\Box PROP2$ the abstraction rule requires that the variable is in rightmost position in the left part (Γ_T) of the context, whereas in $\lambda\Box PROP2$ the rule requires the variable to be in rightmost position with respect to the entire context. To reconcile the abstraction rules we use the Strong Permutation Lemma:

$M \equiv \lambda x_i : Prop.M_1$
$\Gamma_1;\Delta_1 \boxdot \ldots \boxdot \Gamma_n;\Delta_n \vdash \lambda x_i : Prop.M_1 : \forall x : Prop.\psi$, then by Stripping
$\Gamma_1;\Delta_1 \boxdot \ldots \boxdot \Gamma_n, x_i : Prop;\Delta_n \vdash M_1 : \psi$.
Hence by IH (1) $(\Gamma_1;\Delta_1 \boxdot \ldots \boxdot \Gamma_{n-1};\Delta_{n-1})^\sharp \boxdot (\Gamma_n)^\sharp, (x_i)^\sharp : Prop, (\Delta_n)^\sharp \vdash (M_1)^\sharp : \psi^\sharp$.

Since we know (by the condition on the $\forall$-I rule) that
$x_i \notin FV(\Delta_n)$, we have that $(x_i)^\sharp \notin FV(\Delta_n)^\sharp$. This means that the Strong Permutation Lemma for $\lambda\Box PROP2$ can be used to push $(x_i : Prop)$ 'through' $(\Delta_n)^\sharp$, permuting it one by one with all elements of $(\Delta_n)^\sharp$ which results in
(2) $(\Gamma_1; \Delta_1 \boxtimes \ldots \boxtimes \Gamma_{n-1}; \Delta_{n-1})^\sharp \boxtimes (\Gamma_n)^\sharp, (\Delta_n)^\sharp, (x_i)^\sharp : Prop \vdash (M_1)^\sharp : (\psi)^\sharp$.
From $\Gamma_1; \Delta_1 \boxtimes \ldots \boxtimes \Gamma_n, x_i : Prop; \Delta_n \vdash M_1 : \psi$,we also have by Well-typedness that (3) $\Gamma_1; \Delta_1 \boxtimes \ldots \boxtimes \Gamma_n, x_i : Prop \vdash \psi : Prop$ from which we can prove ($\forall$-case, part 1 of this theorem), (4) $(\Gamma_1; \Delta_1 \boxtimes \ldots \boxtimes \Gamma_{n-1}; \Delta_{n-1})^\sharp \boxtimes (\Gamma_n)^\sharp \vdash (\Pi x^\sharp : Prop.\psi^\sharp) : Prop$.
Hence by Thinning
(5) $(\Gamma_1; \Delta_1 \boxtimes \ldots \boxtimes \Gamma_{n-1}; \Delta_{n-1})^\sharp \boxtimes (\Gamma_n)^\sharp, (\Delta_n)^\sharp \vdash (\Pi x^\sharp : Prop.\psi^\sharp) : Prop$. and so by *Abstraction* on (1) and (5)
$(\Gamma_1; \Delta_1 \boxtimes \ldots \boxtimes \Gamma_{n-1}; \Delta_{n-1})^\sharp \boxtimes (\Gamma_n)^\sharp, (\Delta_n)^\sharp \vdash \lambda(x_i)^\sharp : Prop.(M_1)^\sharp : \Pi x^\sharp : Prop.(\psi)^\sharp$
By definition of $\sharp$, $(\Gamma_1; \Delta_1 \boxtimes \ldots \boxtimes \Gamma_n; \Delta_n)^\sharp \vdash (\lambda x_i : Prop.M_1)^\sharp : (\forall x : Prop.\psi)^\sharp$

2.5.8. THEOREM. *For all terms M in $\lambda\Box PROP2$:*

1 *If* $\Gamma_T^1 \boxtimes \ldots \boxtimes \Gamma_T^n \vdash M : Prop$, *then* $(\Gamma_T^1 \boxtimes \ldots \boxtimes \Gamma_T^n)^\flat \vdash M^\flat : Prop$

2 *If* $\Gamma_T^1, \Gamma_P^1 \boxtimes \ldots \boxtimes \Gamma_T^n, \Gamma_P^n \vdash M : \varphi(: Prop)$, *then* $(\Gamma_T^1, \Gamma_P^1 \boxtimes \ldots \boxtimes \Gamma_T^n, \Gamma_P)^\flat \vdash M^\flat : \varphi^\flat$

PROOF. by induction on the structure of the term (M).

In the 'Π-case' of the proof of 1 we need the Strengthening Lemma for $MPTS_\beta$:
For $G_1, x : A, G_2$ a context, and M and B terms, $G_1, x : A, G_2 \vdash M : B$ and $x \notin FV(G_2, M, B) \Rightarrow G_1, G_2 \vdash M : B$.

$M \equiv \Pi x^{Prop} : \varphi.\psi$
$\Gamma_T^1 \boxtimes \ldots \boxtimes \Gamma_T^n \vdash (\Pi x^{Prop} : \varphi.\psi) : Prop$, hence by Stripping
(1) $\Gamma_T^1 \boxtimes \ldots \boxtimes \Gamma_T^n \vdash \varphi : Prop$ and (2) $\Gamma_T^1 \boxtimes \ldots \boxtimes \Gamma_T^n, x^{Prop} : \varphi \vdash \psi : Prop$. Since we know that $x^{Prop} \notin FV(\psi, Prop)$, we have by Strengthening (taking 'G_2'to be empty) on (2) that (3) $\Gamma_T^1 \boxtimes \ldots \boxtimes \Gamma_T^n \vdash \psi^\flat : Prop$. By IH on (1) and (3),
(4) $(\Gamma_T^1 \boxtimes \ldots \boxtimes \Gamma_T^n)^\flat \vdash \varphi^\flat : Prop$ and
(5) $(\Gamma_T^1 \boxtimes \ldots \boxtimes \Gamma_T^n)^\flat, (x^{Prop})^\flat : \varphi^\flat \vdash \psi^\flat : Prop$. Therefore by the $\supset$-rule on (4) and (5) $(\Gamma_T^1 \boxtimes \ldots \boxtimes \Gamma_T^n)^\flat \vdash \varphi^\flat \supset \psi^\flat : Prop$. and hence by definition of $\flat$,
$(\Gamma_T^1 \boxtimes \ldots \boxtimes \Gamma_T^n)^\flat \vdash (\Pi x^{Prop} : \varphi.\psi)^\flat : Prop$.

In the proof of 2, we face again the conflicting requirements of the abstraction rules in the case of abstraction over a propositional variable. As above we solve this using the Strong Permutation Lemma for $\lambda\Box PROP2$. Since we do induction on the structure of the term, rather than on the derivation, we can move the propositional variable to the 'correct' position in the context *before* applying the mapping.

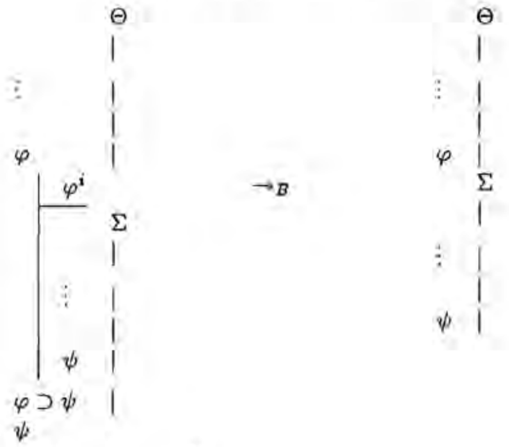
$M \equiv \lambda x_i^{Type^p} : Prop.M_1$
$\Gamma_T^1, \Gamma_P^1 \boxtimes \ldots \boxtimes \Gamma_T^n, \Gamma_P^n \vdash \lambda x_i^{Type^p} : Prop.M_1 : \Pi x : Prop.\psi$, then by Stripping
$\Gamma_T^1, \Gamma_P^1 \boxtimes \ldots \boxtimes \Gamma_T^n, \Gamma_P^n, x_i^{Type^p} : Prop \vdash M_1 : \psi$. Since $x_i^{Type^p} \notin FV\Gamma_P$, we can use the Strong Permutation Lemma for $\lambda PROP2$ to push $(x_i : Prop)$ 'through' Γ_P^n, permuting it one by one 'to the left' with all elements of Γ_P^n which results in
$\Gamma_T^1, \Gamma_P^1 \boxtimes \ldots \boxtimes \Gamma_T^{n-1}, \Gamma_P^{n-1} \boxtimes \Gamma_T^n, x_i^{Type^p} : Prop, \Gamma_P^n \vdash M_1 : \psi$. By IH

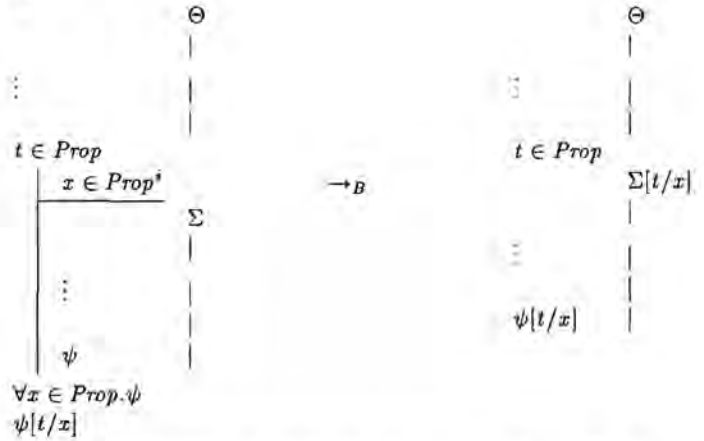
$(\Gamma^1_T, \Gamma^1_P \boxdot \ldots \boxdot \Gamma^{n-1}_T, \Gamma^{n-1}_P)^\flat \boxdot (\Gamma^n_T)^\flat, (x_i^{Type^p})^\flat : Prop; (\Gamma^n_P)^\flat \vdash (M_1)^\flat : (\psi)^\flat$.
By $\forall$-I we get $(\Gamma^1_T, \Gamma^1_P \boxdot \ldots \boxdot \Gamma^{n-1}_T, \Gamma^{n-1}_P)^\flat \boxdot (\Gamma^n_T)^\flat; (\Gamma^n_P)^\flat \vdash$
$\lambda(x_i^{Prop})^\flat : Prop.(M_1)^\flat : \forall(x^{Prop})^\flat : Prop.(\psi)^\flat$. Therefore by the definition of $\flat$,
$(\Gamma^1_T, \Gamma^1_P \boxdot \ldots \boxdot \Gamma^n_T, \Gamma^n_P)^\flat \vdash (\lambda x_i^{Prop} : Prop.M_1)^\flat : (\Pi x^{Prop} : Prop.\psi)^\flat$.

2.6. Proof reductions

It is well-known that for the logics corresponding to the systems in the Logic Cube, cut-elimination corresponds to normalization of β-reduction. The same holds true for the modal logics corresponding to the systems in the Modal Logic Cube. We show this by defining a reduction relation on deductions of $\Box PROP2$.

2.6.1. Definition. The reduction relation $\to_B$ on deductions of $\Box PROP2$ is defined as follows.





In the $\forall$-case, Θ represents the part of the proof preceding the notification $t \in Prop$ (which itself has an atomic proof), and $\Sigma[t/x]$ is the derivation Σ with every occurrence of $x \in Prop^i$

replaced by $t \in Prop$. It is easy to check that if Σ is a derivation of ψ under the hypothesis $x \in Prop$, $\Sigma[t/x]$ is derivation of $\psi[t/x]$.

2.6.2. PROPOSITION. *There is a one-to-one correspondence between reduction steps $\rightarrow_B$ in an A-OK deduction Θ of $\Box PROP2$ and β-reductions in the corresponding proof-term $\Theta^!$ of $\Lambda\Box PROP2$. Hence we have:*

$\rightarrow_B$ is (strongly) normalizing on A-OK deductions of $\Box PROP2$ $\Leftrightarrow$
β-reduction is (strongly) normalizing on proof terms of $\Lambda\Box PROP2$.

PROOF. Immediate from the one-to-one-correspondence between equivalence classes of A-OK proofs and proof terms.

The $\rightarrow_B$ reduction on deductions is a 'propositional' reduction, it removes pairs of introduction and elimination rule applications for $\supset$ and $\forall$. Given this relationship between operations on terms and operations on propositional proofs the question arises if the extended type theoretical system allows for some sort of 'modal proof reduction'.

2.6.1. Reduction in K

First we shall look at the basic modal rules, K-import and K-export. Because of the symmetry between import and 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. It 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.

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

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 is pointless in a natural deduction proof.

2.6.3. DEFINITION. *$\hat{k}\check{k}$ reduction* : $\hat{k}(\check{k}M) \Rightarrow M$

In combination with the mappings '!' and '?', $\hat{k}\check{k}$-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 summetry of the basic modal rules, it is not surprising that we can make a similar observation about sequences of K-export and 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.

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 ($degree(A) = (i, 0)$, for some i).

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

2.6.4. DEFINITION. *$\check{k}\hat{k}$ reduction* : $\check{k}(\hat{k}M) \Rightarrow M$

We shall call both kinds of reduction 'annihilation'; any time a $\check{k}$-function meets a $\hat{k}$-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. $\hat{k}(\check{k}M)$) may always be replaced (by M), regardless of the structure of the term in which it appears[3] (an application $N(\hat{k}(\check{k}M))$ for instance).

The annihilations are presented here as 'structural' reductions, but similar reductions arise by intepreting the modal operator '□' explicitly as a universal quantifier over worlds. A famous example of this approach is Gallin's two sorted theory of types Ty2 ([Gallin 1975]). Instead of structured contexts and import/export-rules, this system has an additional sort of variables allowing direct reference to worlds. Using these 'world variables' (for which we provisionally write w_i (of type W)), the rules for introduction and elimination of modal types are analogous to ∀-intro and ∀-elim. Hence from the 'two sorted' point of view, the annihilations look like this:

$\check{k}(\hat{k}M) : A \Rightarrow M : A$ becomes $(\lambda w_i : W.M)w_i : A \Rightarrow M : A$

$\hat{k}(\check{k}M) : \Box A \Rightarrow M : \Box A$ becomes $\lambda w_i : W(M(w_i)) : \forall w : W.A \Rightarrow M : \forall w : W.A$

In other words, *$\check{k}\hat{k}$ reduction* corresponds to β-reduction and *$\hat{k}\check{k}$ reduction* to η-reduction. However in Ty2 the second reduction is not generally correct, since the condition on η-reduction ($w_i \notin FV(M)$) is not always met.

[3]See [Barendregt 1992].

Despite these remsemblances between the 'structural' and the 'quantifier' view on annihilations, it seems to be easier to generalize the idea of annihilations to modal systems above K in the structural perspective. There simply we can simply look at the effects of combinations of import and export rules, whereas the quantificational view requires that we use several kinds of world variables ('K-worlds', '4-worlds' etc.) or work with second order quantification over a structured collection of worlds ([De Queiroz and Gabbay 1995]). To give an idea of the possibilities for modal proof reduction offered by our type theoretical formalism we will discuss the following topics:

- Other annihilations: besides sequences of K-import and K-export there may be more pointless combinations of an import and an export step in the various normal modal systems.

- Distribution: if the modal functions do not interfere with the operations of typed λ-calculus there could be functions distributing over application and abstraction.

- Other rules: besides annihilation there may be other forms of reduction, perhaps involving groups of functions.

2.6.2. Other annihilations

To find other pointless combinations of import and export steps one can simply check all cases, but apparently these combinations consist of an 'extra' import rule and an 'extra' export rule. Looking at the export rules we see that the D-export rule 'adds $\neg\Box\neg$' to the formula it exports, since there is no import rule that 'subtracts $\neg\Box\neg$' from the formula it imports there can be no pointless combinations involving the D-export rule. The T-export rule leaves every formula it exports unchanged, therefore any combination with an import rule that leaves formulas of a certain form unchanged will result in a detour. We have seen two such import rules: the 4-import rule and the 5-import rule.

Sequences of 4-import followed by immediate T-export and T-export directly followed by 4-import are pointless, they can be eliminated, leaving the rest of the proof to be carried out in exactly the same way as before:

This observation can be formalized type theoretically by means of the following rules:

2.6.5. DEFINITION. $\hat{t}\check{4}$ *reduction* : $\hat{t}(\check{4}M) \Rightarrow M$ and $\check{4}\hat{t}$ *reduction* : $\check{4}(\hat{t}M) \Rightarrow M$.

In the same way reduction rules rules can be given for the combination of 5-import and T-export:

2.6.6. DEFINITION. $\hat{t}\check{5}$ *reduction* : $\hat{t}(\check{5}M) \Rightarrow M$ and $\check{5}\hat{t}$ *reduction* : $\check{5}(\hat{t}M) \Rightarrow M$

We call $\Lambda\Box PROP2$ with all of the annihilations rules added to it $\Lambda\Box PROP2_{\beta,annih}$. In this system we can prove (cf. chapter 3) that the annihilations have all the usual reduction properties:

> **Subject Reduction** If $G \vdash M : A$ and M reduces to M' through a number of annihilations (and β-steps), then $G \vdash M' : A$: 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.

2.6.3. Distribution

Sofar we have only looked at reductions that correspond to 'local' simplifications of deduction proofs; the annihilations remove pointless combinations of *consecutive* modal steps. This raises the question whether applications of import and export rules can be permuted with applications of non-modal rules. If so, more global reductions become possible: originally distant applications of import and export can be brought together and then annihilated.

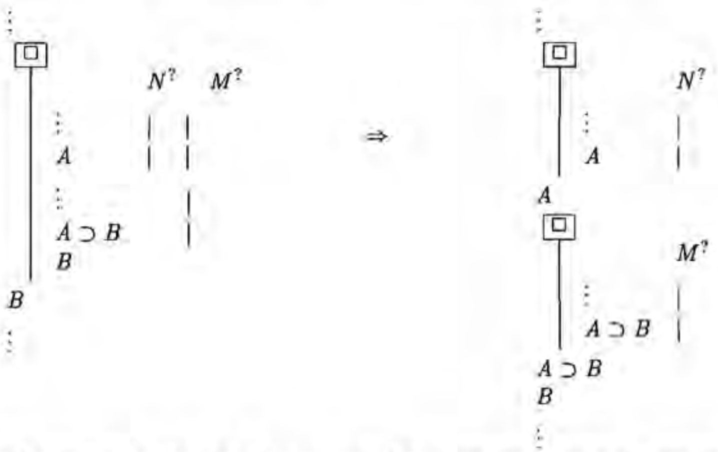
In type theoretical terms, the question of permutation translates as follows: can any of the modal functions $(\check{k}, \hat{4}, \check{5}, \check{b}, \hat{k}, \hat{d}, \hat{t})$ be made to distribute over application and abstraction ? The only candidates for such distributive behaviour are those import and export functions that belong to rules that do not change the types of the statements to which they are applied. The rules for application ($\supset$-elim, $\forall$-elim) and abstraction ($\supset$-intro, $\forall$-intro) demand a certain relation between the form of the types of their premisses. If this relation is not preserved by the modal rule, applications or abstractions possible before the import or export of the premisses may become impossible afterwards. For example, given that $G \vdash N : A$ and

$G \vdash M : A \supset B$ we can apply $\supset$-elim to obtain $G \vdash MN : B$, however after B-importing both premisses ($G \boxdot \varepsilon; \varepsilon \vdash \breve{b}N : \neg\Box\neg A$ and $G \boxdot \varepsilon; \varepsilon \vdash \breve{b}M : \neg\Box\neg(A \supset B)$) $\supset$-elim can no longer be applied.

This observation leaves us with candidates $\breve{4}$, $\breve{5}$, and $\hat{t}$. Even though the 4- and 5-import rule do not change the form of the type, they require a type of a certain form ($\Box\varphi$ and $\neg\Box\varphi$, respectively) which precludes permutation with any of the rules $\supset$-intro, $\supset$-elim, $\forall$-intro, and $\forall$-elim. Hence we now investigate whether $\hat{t}$ can be made to distribute over application and abstraction.

Distribution over application

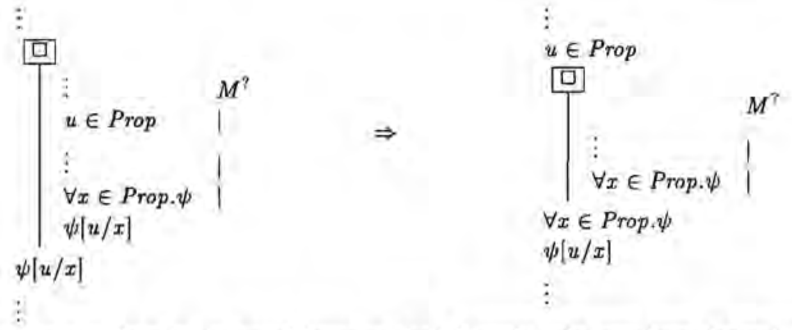
A straightforward attempt to formalize the possible distribution of $\hat{t}$ over application is the rule $\hat{t}(MN) \Rightarrow (\hat{t}M)(\hat{t}M)$. For proof terms M and N, this rule gives a correct procedure to permute applications of $\supset$-elim and T-export. Given that $G \boxdot \varepsilon; \varepsilon \vdash M : A \supset B$ and $G \boxdot \varepsilon; \varepsilon \vdash N : A$, where first $\supset$-elim is applied ($G \boxdot \varepsilon; \varepsilon \vdash MN : B$) and then T-export ($G \vdash \hat{t}(MN) : B$), we could just as well have T-exported M and N first ($G \vdash \hat{t}M : A \supset B$ and $G \vdash \hat{t}N : A$), and then applied $\supset$-elim ($G \boxdot \varepsilon; \varepsilon \vdash (\hat{t}M)(\hat{t}N) : B$). The following figure shows that the corresponding reduction on deduction proofs splits the modal subordinate proof of B ending in T-export into two shorter modal subordinate proofs ending in T-export.



Clearly, the derivation on the right is a correct deduction proof if the derivation on the left is.

In the other possible application case M is a proof term, N is an object term, and the reduction rule should correctly describe the permutation of applications of $\forall$-elim and T-export. However, it is easy to see that this is not the case. Suppose that $G \boxdot \varepsilon; \varepsilon \vdash M : \forall x : Prop.\psi$ (where $G \equiv \Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n$) and $F \boxdot \varepsilon \vdash u : Prop$ (where $F \equiv \Gamma_1 \boxdot \ldots \boxdot \Gamma_n$). Then $\forall$-elim yields $G \boxdot \varepsilon; \varepsilon \vdash Mu : \psi[u/x]$, and hence by T-export we have $G \vdash \hat{t}(Mu) : \psi[u/x]$. Starting from the same situation, we cannot first apply T-export and then $\forall$-elim, since $u : Prop$ is not a proof term and hence T-export cannot be applied to it.

Looking at the corresponding deduction helps to find a correct distribution rule for $\forall$-elim.



The notification $u \in Prop$ in the proof figure on the left could only have been derived in two ways: by reiterating a hypothesis $u \in Prop$ (where u is an object variable) or by the *term*-rule. Since $u \in Prop$ lies in the modal interval it cannot be the reiteration of a hypothesis, as reiteration across modal intervals is not allowed. Hence $u \in Prop$ is derived by the term rule and has an atomic proof. Therefore $u \in Prop$ could just as well have been derived in the main proof right before the start of the modal proof. After proving $\forall x \in Prop.\psi$ by $M^?$ in the subordinate proof, $\forall x \in Prop.\psi$ can be brought to the main proof where $\forall$-elim can then be applied as in the right proof figure above.

Type theoretically, this reduction corresponds to the rule $\hat{\imath}(MN) \Rightarrow (\hat{\imath}M)N$ where M is a proof term and N is an object term. At first glance, this reduction rule may seem incorrect, since the rules of $\Lambda\Box PROP2$ do not allow us to derive $G \vdash (\hat{\imath}M)u : \psi[u/x]$ from $G \boxdot \varepsilon; \varepsilon \vdash M : \forall x : Prop.\psi$ and $F \boxdot \varepsilon \vdash u : Prop$. The difficulty, obviously, is to derive that $F \vdash u : Prop$ given that $F \boxdot \varepsilon \vdash u : Prop$. In chapter 3 we will show that this "backwards transfer" of non-proof terms is actually a derived rule of the MPTSs. For $\Lambda\Box PROP2$, the rule looks like this

$$refsnart\ \frac{F \boxdot \varepsilon \vdash \varphi : Prop}{F \vdash \varphi : Prop}.$$

Given *refsnart* it is immediate that the proposed reduction rule is correct.

2.6.7. Definition. **$\hat{\imath}$-distribution over application**

For M and N proof terms: $\hat{\imath}(MN) \Rightarrow (\hat{\imath}M)(\hat{\imath}N)$

For M a proof term and N an object term: $\hat{\imath}(MN) \Rightarrow (\hat{\imath}M)N$

Note that the "import pseudo terms" $(\check{k}T, \check{4}T, \check{5}T, \check{b}T)$ are not applications, hence T-export will not distribute over an import rule, e.g. $\hat{\imath}(\check{k}M) \not\Rightarrow (\hat{\imath}\check{k})(\hat{\imath}M)$.

Distribution over abstraction

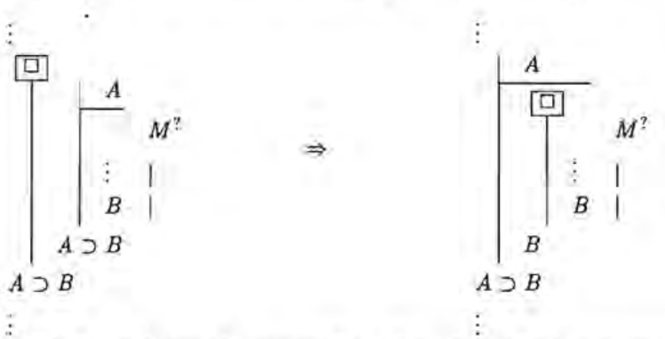
The simplest way in which $\hat{\imath}$ could distribute over abstraction is expressed by the following rule.

2.6.8. Definition. **$\hat{\imath}$-distribution over abstraction**

$\hat{\imath}(\lambda x : Prop.M) \Rightarrow \lambda x : Prop.(\hat{\imath}M)$

$\hat{\imath}(\lambda p : \varphi.M) \Rightarrow \lambda p : \varphi.(\hat{\imath}M)$

This rule claims that it makes no difference whether one first abstracts a variable (say $p : A$) over a term $(M : B)$ in the subordinate context $(G \boxdot \varepsilon;\varepsilon \;\vdash\; \lambda p : A.M : A \supset B)$ and then T-exports the resulting statement $(G \;\vdash \lambda p : A.M : A \supset B)$, or first T-exports the term to a context extended with $p : A$ $(G, p : A \;\vdash\; \hat{\imath}M : B)$ and then abstracts $(G \;\vdash\; \lambda p : A.(\hat{\imath}M) : A \supset B)$. The correctness of this rule depends on the occurrence of the abstraction variable in the term $(p \in FV(M))$; since abstraction corresponds to the discharge a hypothesis in natural deduction (∀-intro, ⊃-intro), the effect of the rule is to move a hypothesis(-interval) of the strict subordinate proof to the main proof:



Reducing the left derivation to the one on the right only results in a correct deduction proof if the hypothesis A is not used in the proof $M^?$ of B. Hence *$\hat{\imath}$ distribution over abstraction* has to be restricted to cases of 'vacuous' abstraction $(x_i, p_i \notin FV(M))$ to obtain a correct rule. Although we cannot distribute $\hat{\imath}$ over abstractions correctly in single steps, we do not have to give up the idea of permuting T-export with non-modal rules. The distribution rule given above together with *$\hat{\imath}$ distribution over application* and one new rule (to be discussed below) allow us to bring terms of the form $\hat{\imath}M$ in a sort of '$\hat{\imath}$-normal form' which is again an inhabitant of the type of $\hat{\imath}M$.

The combined effect of the rules defined so far is that $\hat{\imath}$ can be 'moved through the term' by repeated distribution over applications and abstractions. This movement is stopped when $\hat{\imath}$ meets

i a variable: x_i or p_i,

ii an import subterm: $\check{k}N, \check{4}N, \check{5}N, \check{b}N$,

iii an export subterm: $\hat{k}N, \hat{d}N, \hat{\imath}N$.

A variable occurring as a subterm of $(\hat{\imath}M)$ corresponds to the reiteration of a hypothesis of the strict subordinate proof. Since all hypotheses of a strict subordinate proof have to be discharged before T-export is applied, the variables occurring as subterms in $\hat{\imath}M$ are bound. We already saw that *$\hat{\imath}$ distribution over abstraction* moves the hypotheses of the strict subordinate proof to the main proof, hence we now propose a rule that does the same for the reiterations of these hypotheses.

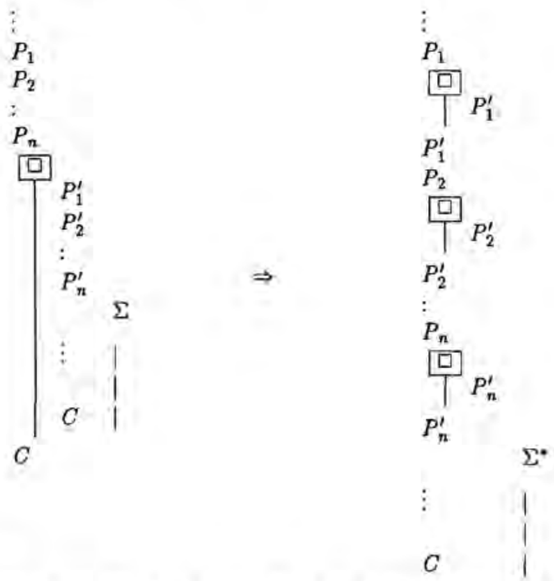
2.6.9. DEFINITION. **$\hat{\imath}$ *var*-reduction**

for all object variables x, $\hat{\imath}x : Prop \;\Rightarrow\; x : Prop$

for all proof variables p, $\hat{t}p : \varphi \;\Rightarrow\; p : \varphi$

When no more applications of any of the three distribution rules are possible, a term is in '$\hat{t}$-normal form'. Using the above case distinction, we know that $\hat{t}$s in a term in $\hat{t}$-normal form occur in front of an import subterm or an export subterm (cases *ii* and *iii*). A $\hat{t}$ in front of an import subterm signifies a 'minimal strict subordinate proof': the application of some import rule immediately followed by T-export. These minimal proofs offer possibilities for further reduction, when the import rule is 4 or 5-import it can be eliminated by annihilation.

In the following schematic example we illustrate the effect of reduction to $\hat{t}$-normal form on natural deduction proofs.



In the proof figure on the left, C is derived in a strict subordinate proof ending in T-export. The derivation Σ of C uses the formulas $P'_1, P'_2, \ldots, P'_n$, which result from importing $P_1, P_2, \ldots, P_n$ occurring in the main proof. The proof figure on the right shows the effect of the distribution of T-export through Σ. The strict subordinate proof ending in T-export is reduced to a number of minimal strict subordinate proofs ending in T-export, which serve only to create occurrences of $P'_1, P'_2, \ldots, P'_n$ in the main proof. Once this is achieved C can be derived in the main proof in the same way as it was the derived in the strict subordinate proof. The only possible difference between Σ and Σ^* is that the hypothesis-intervals lying inside Σ now range from just above the occurrence of P_1 down into Σ^*. In terms of possible worlds this proof transformation can be understood in the following way: since 'T-worlds' are accessible to themselves, a proof in a reachable world of a T-world could just as well have been performed in the T-world itself.

The nice reduction properties that hold for the annihilations can be proved for distribution of $\hat{t}$ over application, but we can already see from the congruence cases for distribution over abstraction that these properties are difficult to prove for the full set of distribution rules. New β-redexes can emerge while distributing over abstraction, making the proof of Strong

Normalisation complicated. Subject Reduction is lost for distribution over abstractions since (using the mappings) terms corresponding to 'illegal deduction proofs' will result. However, the set of distribution rules as a whole can be used to simplify modal proofs (see the example in 2.6.6). Therefore we propose the following.

2.6.10. CONJECTURE.

> **Weak Normalisation** *The set of annihilation rules together with the $\hat{t}$-distribution rules is weakly normalizing: there is a procedure which turns terms in to $\hat{t}$-normal forms.*
>
> **Subject Reduction for $\hat{t}$-normal forms** *Given terms M and M' where M' is a $\hat{t}$-normal form of M, if $G \vdash M : \varphi$ then $G \vdash M' : \varphi$: the $\hat{t}$-normal form is again a proof of the original proposition.*

2.6.4. Other Rules

In the previous sections we discussed the annihilating function pairs and distributing functions for the modal logics we are concerned with. However, there may be other possible reductions if we take reduction to be a relation between sequences of functions (annihilation being the reduction of a pair of functions to the empty sequence).

One way to look for such reductions is to concentrate on the inclusions between the various normal modal logics. Among these inclusions, those that do not arise from the mere addition of a rule (like $KT \subset KT4$) are interesting, because the stronger system can prove all theorems of the weaker system with a different set of rules. This means that certain sequences of steps in proofs of the stronger system are equivalent to a sequence of steps in proofs of the weaker system. Under the propositions-as-types-interpretation of modal logic these relations between sequences of steps may turn out to be formalizable as reduction rules on the import and export functions appearing in terms representing these proofs.

One of the inclusions mentioned in [Chellas 1980] is $KDB \subset KTB$; theorems of KDB which were proved using the D-export rule can be proved in KTB using the T-export rule instead. A little doodling shows that any sequence of T-export, B-import, and T-export in KTB corresponds to an application of D-export in KDB:

This relation could be formalized by means of the reduction rule: $\hat{t}(\check{b}(\hat{t}M)) \Rightarrow \hat{d}M$. Interesting inclusions can also yield more complex relations, e.g. from the inclusion $KB4 \subset KT5$ we can conclude that the work of the $\check{b}$-function can be taken over by the $\check{k}$, $\bar{5}$, $\hat{k}$, and $\hat{t}$ functions. Quite a bit of doodling shows that $\check{b}M : \neg\Box\neg A$ in $KB4$ (resulting from an application of B-import to $M : A$) corresponds to $\breve{5}(\lambda y : \Box\neg A.(\hat{t}(\check{k}y))M : \neg\Box\neg A$ (where $M : A$) in $KT5$. Obviously such a relation cannot be formalized as a reduction on sequences of functions, it requires major surgery on terms.

These examples suggest that the possibilities of finding 'other rules' may be limited if we require that they are reductions between sequences of functions. However, there is a clear motivation for this requirement: reductions between sequences of functions do not affect the independence of the 'modal operations' and the usual operations of typed λ-calculus. When other sorts of reductions are added to the type sytem this orthogonality may be lost, making it much more difficult to ascertain that the system is well-behaved.

2.6.5. Reduction rules for the extension by axioms

In view of the equivalence of the 'extension by rules' and the 'extension by axioms', the reader may wonder why no reduction rules for the latter extension were given. The reason for this is not that there are no pointless combinations of steps in axiomatic proofs, as can be seen from the following example: first $\Box\Box A$ is derived from $\Box A$ and an A-instance of the 4-axiomschema, then $\Box A$ is in turn derived from $\Box\Box A$ by means of a $\Box A$-instance of the T-axiomschema.

$$
\begin{array}{lcl}
\vdots & & \\
\Box A & & \\
\forall \alpha \in Prop.(\Box\alpha \supset \Box\Box\alpha)) & & \vdots \\
A \in Prop & & \Box A \\
\Box A \supset \Box\Box A & \Rightarrow & \\
\Box\Box A & & \vdots \\
\forall \alpha \in Prop.(\Box\alpha \supset \alpha)) & & \\
\Box\Box A \supset \Box A & & \\
\Box A & & \\
\vdots & & \\
\quad (c_t(\Box A))(c_4 AM) & & M
\end{array}
$$

Assuming that the 4- and T-axiomschemas are in $\mathcal{A}^{Logic}$ (inhabited by c_4 and c_t respectively), we can try to extract an axiomatic version of $\tilde{t}\tilde{4}$ *reduction* from this example in the form of a rule like $(c_t(\Box A))(c_4 AM) \Rightarrow M$. However, such a rule is not general enough; the reduction should hold for any proposition of the form $\Box\varphi$, not just for $\Box A$. To express this, the rule will probably have to be to stated in the form of a schema: $(c_t(\Box\alpha))(c_4\alpha M) \Rightarrow M$ (where the type of M is of the general form $\Box\varphi$ and $\alpha = \Box\varphi$). This shows that reduction rules for the extension by axioms have a format that is rather different from that of traditional reduction rules.

2.6.11. EXAMPLE. To illustrate the effect of the reductions we shall now take a spectacularly inefficient proof of something trivial in $KT45$:

1.	$\Box\Box\Box A$	
2.	$\Box A \supset \bot$	
3.	$\Box\Box\Box A$	(reiteration 1)
4.	[$\Box$] $\Box\Box A$	(K-import 3)
5.	$\Box\Box A$	
6.	$\Box\Box A$	(reiteration 5)
7.	$\Box\Box A \supset \Box\Box A$	($\supset$-intro 5-6)
8.	$\Box\Box A$	($\supset$-elim 4,7)
9.	$\Box\Box A$	(T-export 8)
10.	[$\Box$] $\Box A$	(K-import 9)
11.	$\Box A$	(T-export 10)
12.	[$\Box$] $\Box A \supset \bot$	(5-import 2)
13.	$\Box A$	(4-import 12)
14.	$\bot$	($\supset$-elim 12,13)
15.	$\Box\bot$	(K-export 14)
16.	[$\Box$] $\bot$	(K-import 15)
17.	$\bot$	(T-export 16)
18.	$(\Box A \supset \bot) \supset \bot$	($\supset$-intro 2-17)
19.	$\Box\Box\Box A \supset ((\Box A \supset \bot) \supset \bot)$	($\supset$-intro 1-18)

Using '!', this natural deduction proof can be mapped to the term
$\lambda x : \Box\Box\Box A.(\lambda z : \Box A \supset \bot.\hat{t}(\check{k}(\hat{k}((\breve{5}z)(\check{4}(\hat{t}(\check{k}(\hat{t}((\lambda y : \Box\Box A.y)(\check{k}x))))))))))$,
which can then be simplified by means of some $\Rightarrow$- and β-reductions:

$$\lambda x : \Box\Box\Box A.(\lambda z : \Box A \supset \bot.\hat{t}(\check{k}(\hat{k}(\breve{5}z)(\check{4}(\hat{t}(\check{k}(\hat{t}((\lambda y : \Box\Box A.y)(\check{k}x))))))))) \overset{(\check{k}\hat{k}\ red.)}{\Longrightarrow}$$
$$\lambda x : \Box\Box\Box A.(\lambda z : \Box A \supset \bot.\hat{t}((\breve{5}z)(\check{4}(\hat{t}(\check{k}(\hat{t}((\lambda y : \Box\Box A.y)(\check{k}x)))))))) \overset{(\hat{t}\ distr.appl.)}{\Longrightarrow}$$
$$\lambda x : \Box\Box\Box A.(\lambda z : \Box A \supset \bot.(\hat{t}(\breve{5}z))(\hat{t}(\check{4}(\hat{t}(\check{k}(\hat{t}((\lambda y : \Box\Box A.y)(\check{k}x)))))))) \overset{(\hat{t}\breve{5}\ red.)}{\Longrightarrow}$$
$$\lambda x : \Box\Box\Box A.(\lambda z : \Box A \supset \bot.(z(\hat{t}(\check{4}(\hat{t}(\check{k}(\hat{t}((\lambda y : \Box\Box A.y)(\check{k}x))))))))) \overset{(\hat{t}\check{4}\ red.)}{\Longrightarrow}$$
$$\lambda x : \Box\Box\Box A.(\lambda z : \Box A \supset \bot.(z(\hat{t}(\check{k}(\hat{t}((\lambda y : \Box\Box A.y)(\check{k}x))))))) \overset{(\hat{t}\ distr.appl.)}{\Longrightarrow}$$
$$\lambda x : \Box\Box\Box A.(\lambda z : \Box A \supset \bot.(z(\hat{t}(\check{k}(\hat{t}(\lambda y : \Box\Box A.y)(\hat{t}(\check{k}x))))))) \overset{(\hat{t}\ distr.abstr.)}{\Longrightarrow}$$
$$\lambda x : \Box\Box\Box A.(\lambda z : \Box A \supset \bot.(z(\hat{t}\check{k}((\lambda y : \Box\Box A.\hat{t}y)(\hat{t}(\check{k}x)))))) \overset{(\hat{t}\ varred.)}{\Longrightarrow}$$
$$\lambda x : \Box\Box\Box A.(\lambda z : \Box A \supset \bot.(z(\hat{t}\check{k}((\lambda y : \Box\Box A.y)\hat{t}(\check{k}x))))) \overset{(\beta\ red.)}{\Longrightarrow}$$
$$\lambda x : \Box\Box\Box A.(\lambda z : \Box A \supset \bot.(z(\hat{t}(\check{k}(\hat{t}(\check{k}x))))))$$

Mapping the simplified term back to a natural deduction proof (using '?'), yields a reasonably smart proof of $\Box\Box\Box A \supset ((\Box A \supset \bot) \supset \bot)$ in KT:

1.	$\Box\Box\Box A$	
2.	$\Box A \supset \bot$	
3.	$\Box\Box\Box A$	(reiteration 1)
4.	$\boxed{\Box}$ $\Box\Box A$	(K-import 3)
5.	$\Box\Box A$	(T-export 4)
6.	$\boxed{\Box}$ $\Box A$	(K-import 5)
7.	$\Box A$	(T-export 6)
8.	$\bot$	($\supset$-elim 2,7)
9.	$(\Box A \supset \bot) \supset \bot$	($\supset$-intro 2-8)
10.	$\Box\Box\Box A \supset ((\Box A \supset \bot) \supset \bot)$	($\supset$-intro 1-9)

Chapter 3

Meta theory of MPTSs

Modal Pure Type Systems (MPTSs) are an extension of Pure Type Systems, proposed with the aim of giving a 'propositions-as-types'-interpretation of modal propositional logics as well as predicate logics. In this chapter we investigate the meta theory of the MPTSs introduced in chapter 1.

First we briefly remind the reader of the definition of the Modal Logic Cube of MPTS's, and define some notions needed for the meta theoretical proofs. After some preliminary results in the second section we will show that the meta theoretical properties that hold for the systems in the Logic Cube continue to hold for the systems on the Modal Logic Cube. The proofs of the main properties are more or less analoguous to the proofs of these properties for PTS's. The difficulty is to prove a sufficiently strong Stripping Lemma, as is explained in section 3. The chapter ends with a discussion of the standard 'rewrite properties' for the modal reduction rules defined in the previous chapter.

3.1. The Modal Logic Cube

3.1.1. DEFINITION. **Pseudoterms** For $\mathcal{S}$ some set, the set $\mathcal{T}$ of pseudoterms over $\mathcal{S}$ is

$$\mathcal{T} ::= \mathcal{S} | \mathit{Var} | \Pi\, \mathit{Var} : \mathcal{T}.\mathcal{T}) | (\lambda\, \mathit{Var} : \mathcal{T}.\mathcal{T}) | \mathcal{T}\mathcal{T} | \Box\mathcal{T} | \check{k}\mathcal{T} | \check{4}\mathcal{T} | \check{5}\mathcal{T} | \check{b}\mathcal{T} | \hat{k}\mathcal{T} | \hat{d}\mathcal{T} | \hat{t}\mathcal{T} | \mathcal{C},$$

where *Var* is a countable set of variables, and $\mathcal{C}$ is a countable set of constants. In principle we think of the set *Var* as partitioned over $\mathcal{S}$, that is to say that the variables are indexed with their 'sort' (e.g. $x^{Prop} : A$ where $A : Prop$). However, in many of the following proofs this division of variables is not significant, and in those cases it will be neglected.

Given the additional pseudoterms of the MPTSs over PTSs, the PTS-definition of free variables has to be extended.

3.1.2. DEFINITION. **Free variables** *A map $FV : \mathcal{T} \to P(V)$ is defined as follows:*

(i) $FV(s) = FV(c) = \emptyset$ for all $s \in \mathcal{S}$ and $c \in \mathcal{C}$

(ii) $FV(x) = \{x\}$

(iii) $FV(AB) = FV(A) \cup FV(B)$

(iv) $FV(\lambda x : A.B) = FV(A) \cup (FV(B) - \{x\})$

(v) $FV(\Pi x : A.B) = FV(A) \cup (FV(B) - \{x\})$

(vi) $FV(\Box A) = FV(A)$, $FV(\check{k}A) = FV(\check{4}A) = FV(\check{5}A) = FV(\check{b}A) = FV(A)$,
$FV(\hat{k}A) = FV(\hat{d}A) = FV(\hat{t}A) = FV(A)$.

3.1.3. Definition. **Generalized 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: $x_1 : A_1, \ldots, x_n : A_n$. The empty context is denoted by ε. If
$\Gamma = x_1 : A_1, \ldots, x_n : A_n$ then $\Gamma, x : B = x_1 : A_1, \ldots, x_n : A_n, x : B$.

(iiia) A *generalized pseudo-context* is a finite ordered sequence of pseudo-contexts and separators $\boxdot$, in which all variables occuring as subjects are different: $G = \Gamma_1 \boxdot \ldots \boxdot \Gamma_n$.
If $G = x_1 : A_1, \ldots \boxdot \ldots \boxdot\ x_m : A_m, \ldots, x_n : A_n$
then $G, x : B = x_1 : A_1, \ldots \boxdot \ldots \boxdot x_m : A_m, \ldots, x_n : A_n, x : B$,
and $G \boxdot \varepsilon = x_1 : A_1, \ldots \boxdot \ldots \boxdot x_m : A_m, \ldots, x_n : A_n \boxdot \varepsilon$.

(iiib) Alternatively an *inductive definition* of the set of generalized pseudo contexts $\mathcal{G}$ can be given based on the set $\mathcal{O}$ of PTS-contexts:
$\mathcal{G} \ = \ \mathcal{O} \mid \mathcal{G} \boxdot \mathcal{O}$

We take $\Gamma, \Gamma', \ldots$ to be ranging over $\mathcal{O}$, and $G, G', \ldots$ to range $\mathcal{G}$.'ε', the empty context, denotes the empty sequence of statements. It is introduced to clarify the notation of the modal rules.

For generalized pseudo-contexts the map FV (or Dom) is defined in stages:

(i) $FV(\varepsilon) = \emptyset$

(ii) $FV(\Gamma, x : A) = FV(\Gamma) \cup \{x\}$

(iii) $FV(G \boxdot \Gamma) = FV(G) \cup FV(\Gamma)$

Based on the inclusion relation between PTS-contexts, we define two inclusion relations for generalized contexts.

3.1.4. Definition. **Context inclusion**
For Γ and Γ' PTS-pseudo-contexts, and G and G' generalized pseudo-contexts:

(i) Γ *is part of* Γ', $\Gamma \subseteq \Gamma'$, if every declaration $x : A$ in Γ is also in Γ'.

(ii) G is *part of* G', $G \subseteq G'$, if $G \equiv \Gamma_1 \boxdot \ldots \boxdot \Gamma_n$ and $G' \equiv \Gamma'_1 \boxdot \ldots \boxdot \Gamma'_n$ for some $n \in \mathbb{N}$ and $\forall i(1 \leq i \leq n)(\Gamma_i \subseteq \Gamma'_i)$ as under (i).

(iii) G is an *initial part* of G', $G \leq G'$, if $G \equiv \Gamma_1 \boxdot \ldots \boxdot \Gamma_m$ and $G' \equiv \Gamma'_1 \boxdot \ldots \boxdot \Gamma'_n$ for some $n, m \in \mathbb{N}$, where $m \leq n$ and $\forall i(1 \leq i \leq m)(\Gamma_i \subseteq \Gamma'_i)$.

For $G \subseteq G'$, we require G and G' to be of the same 'modal depth'; this subset relation for generalized contexts can be seen as taking the sum of the subset relation over all pairs of corresponding 'simple' contexts in G and G'. Note that the *part of*-relation is a special case of the *intial part of*-relation (for $n = m$ the definitions are equivalent), that has no counterpart in PTSs.

3.1.5. DEFINITION. **Modal Pure Type Systems**
A *Modal Pure Type System with β-conversion* $MPTS_\beta$ is given by a set $\mathcal{S}$ of *sorts* containing *Prop*,*Set*, and *Type*, a set $\mathcal{A}^{Type} \subset \mathcal{S} \times \mathcal{S}$ of *typing axioms*, a set $\mathcal{A}^{Logic} \subset \mathcal{C} \times \mathcal{T}$ 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 $\Box\lambda_\beta(\mathcal{S}, \mathcal{A}, \mathcal{R})$ and is the typed λ-calculus with the following deduction rules[1]:

$$(axiom)\quad \varepsilon \vdash s_1 : s_2 \quad \text{if } s_1 : s_2 \in \mathcal{A}^{type} \qquad \varepsilon \vdash c : A : Prop \quad \text{if } c : A \in \mathcal{A}^{Logic}$$

$$(start)\quad \frac{G \vdash A : s}{G,\ x : A \vdash x : A}$$

$$(weakening)\quad \frac{G \vdash A : B \quad G \vdash C : s}{G,\ x : C \vdash A : B}$$

$$(product)\quad \frac{G \vdash A : s_1 \quad G,\ x : A \vdash B : s_2}{G \vdash (\Pi x : A.B) : s_3} \quad \textit{if } (s_1, s_2, s_3) \in \mathcal{R}$$

$$(application)\quad \frac{G \vdash F : (\Pi x : A.B) \quad G \vdash a : A}{G \vdash Fa : B[x := a]}$$

$$(abstraction)\quad \frac{G,\ x : A \vdash b : B \quad G \vdash (\Pi x : A.B) : s}{G \vdash (\lambda x : A.b) : (\Pi x : A.B)}$$

$$(conversion)\quad \frac{G \vdash A : B \quad G \vdash B' : s \quad B =_\beta B'}{G \vdash A : B'}$$

$$(boxing)\quad \frac{G \vdash A : Prop}{G \vdash \Box A : Prop}$$

$$(transfer_1)\quad \frac{G \vdash A : s}{G \boxdot \varepsilon \vdash A : s}$$

$$(transfer_2)\quad \frac{G \vdash A : B : Type}{G \boxdot \varepsilon \vdash A : B}$$

$$(transfer_3)\quad \frac{G \vdash A : B : Set}{G \boxdot \varepsilon \vdash A : B}$$

$$(transfer_{ax})\quad \frac{G \vdash c : A : Prop}{G \boxdot \varepsilon \vdash c : A}$$

$$(K\ import)\quad \frac{G \vdash A : \Box B : Prop}{G \boxdot \varepsilon \vdash \check{k}A : B}$$

$$(K\ export)\quad \frac{G \boxdot \varepsilon \vdash A : B : Prop}{G \vdash \hat{k}A : \Box B}$$

where s ranges over the set of sorts $\mathcal{S}$, c over the set of constants $\mathcal{C}$, x ranges over variables, and it is assumed that in the rules (*start*) and (*weakening*) the newly declared variable x is always fresh, that is, it has not yet been declared in G.

[1] Again using the notational convention (cf. chapter 1) that $G \vdash A : B : C$ stands for $G \vdash A : B$ and $G \vdash B : C$.

MPTSs can have various import and export rules, but they have to have the K-import and K-export rule, since these rules hold for all normal modal operators. In this chapter we add all of the additional import and export rules introduced in chapter 1.3 to the MPTSs:

$$4\,import\,\frac{G\ \vdash\ A:\Box B:Prop}{G\boxdot\varepsilon\ \vdash\ \check{4}A:\Box B}\quad 5\,import\,\frac{G\ \vdash\ A:\neg\Box B:Prop}{G\boxdot\varepsilon\ \vdash\ \check{5}A:\neg\Box B}\quad B\,import\,\frac{G\ \vdash\ A:B:Prop}{G\boxdot\varepsilon\ \vdash\ \check{b}A:\neg\Box\neg B}$$

$$D\,export\,\frac{G\boxdot\varepsilon\ \vdash\ A:B:Prop}{G\ \vdash\ \hat{d}A:\neg\Box\neg B}\qquad T\,export\,\frac{G\boxdot\varepsilon\ \vdash\ A:B:Prop}{G\ \vdash\ \hat{t}A:B}$$

This allows us to prove at once the meta theoretical properties of all MPTSs that have a subset of these import and export rules.

3.1.6. DEFINITION. **Modal Logic Cube**
The *cube of modal logical typed lambda calculi* consists of the following eight $MPTS_\beta$s. Each of them has

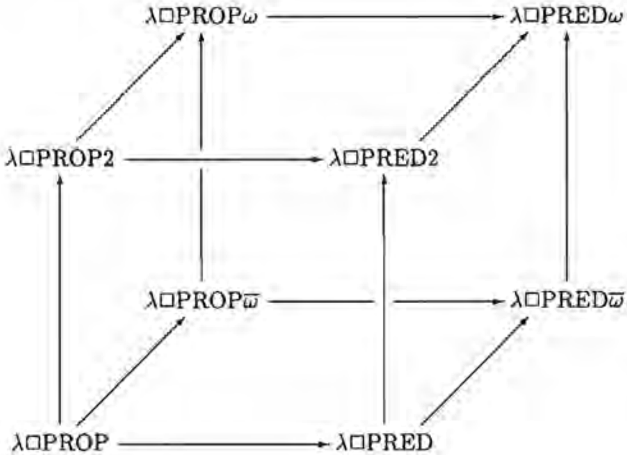
$$\mathcal{S}=\{Prop, Set, Type^p, Type^s\}$$

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

$$\mathcal{A}^{Logic}=c:(\forall x:Prop.((x\supset\bot)\supset\bot)\supset x)$$

The rules of the systems are given by the following list:

$\lambda\Box PROP$	$(Prop, Prop)$
$\lambda\Box PROP2$	$(Prop, Prop)$, $(Type^p, Prop)$
$\lambda\Box PROP\underline{\omega}$	$(Prop, Prop)$, $(Type^p, Type^p)$
$\lambda\Box PROP\omega$	$(Prop, Prop)$, $(Type^p, Prop)$, $(Type^p, Type^p)$
$\lambda\Box PRED$	$(Prop, Prop)$, (Set, Set), $(Set, Prop)$, $(Set, Type^p)$
$\lambda\Box PRED2$	$(Prop, Prop)$, (Set, Set), $(Set, Prop)$, $(Set, Type^p)$, $(Type^p, Prop)$
$\lambda\Box PRED\underline{\omega}$	$(Prop, Prop)$, (Set, Set), $(Set, Prop)$, $(Set, Type^p)$, $(Type^p, Set)$, $(Type^p, Type^p)$
$\lambda\Box PRED\omega$	$(Prop, Prop)$, (Set, Set), $(Set, Prop)$, $(Set, Type^p)$, $(Type^p, Set)$, $(Type^p, Type^p)$, $(Type^p, Prop)$



3.2. Preliminaries

In this section we prove a few of the standard lemmas from the meta theory of PTSs for the modal system. We start with some terminology that is to be used in later proofs.

3.2.1. DEFINITION. **Terminology**
Let G be a generalized pseudo-context and A be a pseudo-term.

(i) G is called *legal* if $\exists P, Q \in \mathcal{T}\ G \vdash P : Q$.

(ii) A is called a *G-term* if $\exists B \in \mathcal{T}(G \vdash A : B$ or $G \vdash B : A)$.

(iii) A is called a *G-type* if $\exists s \in \mathcal{S}(G \vdash A : s)$.

(iv) If $G \vdash A : s$ then A is called a *G-type of sort s*.

(v) A is called a *G-element* if $\exists B \in \mathcal{T} \exists s \in \mathcal{S}(G \vdash A : B : s)$.

(vi) If $G \vdash A : B : s$ then A is called a *G-element of type B and of sort s*.

(vii) $A \in \mathcal{T}$ is called *legal* if $\exists G, B(G \vdash A : B$ or $G \vdash B : A)$.

(viii) $A \in \mathcal{T}$ is called *typable in* $\lambda(\mathcal{S}, \mathcal{A}, \mathcal{R})$ if $\exists G(G \vdash A : B$ or $G \vdash B : A)$ for some B.

(ix) $A \in \mathcal{T}$ is called *typable in G* for G a legal generalized context and A a term, if $G \vdash A : B$ or $G \vdash B : A$ for some B.

First we show that generalized contexts are not that different from the usual pseudocontexts, by proving a Free Variable Lemma, Start Lemma, and Substitution Lemma.

3.2.2. LEMMA. **Free Variable Lemma**
Let $G \equiv x_1 : A_1, \ldots \boxdot \ldots \boxdot \ldots, x_n : A_n$ (or $\Gamma_1 \boxdot \ldots \boxdot \Gamma_m$) be a legal context, say $G \vdash A : B$, then the following hold:

(i) The $x_1, \ldots, x_n$ are all distinct

(ii) $FV(A)$, $FV(B) \subseteq \{x_1, \ldots x_n\}$

(iii) $FV(A_i) \subseteq \{x_1, \ldots, x_{i-1}\}$ for $1 \leq i \leq n$

PROOF. Proof of (i), (ii), and (iii). By induction on the derivation of $G \vdash A : B$

Proof of (i). For the non-modal cases the induction is straightforward. Note that the *axiom* cases are trivial because of the empty context, and that the variables introduced by the *start* and *weakening* rules are fresh and hence distinct from the variables already present in the context.

For the *transfer-*, *import-* and *export*-cases we need the observation that $FV(G \boxdot \varepsilon) = FV(G)$, e. g. :

> Transfer$_1$ $G \vdash A : B$ is $G' \boxdot \varepsilon \vdash C : s$ where $G \equiv G' \boxdot \varepsilon$, an immediate consequence of $G' \vdash C : s$. By IH the $x_1, \ldots, x_n$ in G' are all distinct, and since (by definition) $FV(G' \boxdot \varepsilon) = FV(G')$, the variables in G are all distinct.

Proof of (ii). The interesting cases are the modal cases and those for *axiom* and *abstraction*. Note that all typing axioms as well logical axioms have no free variables. The *abstraction*-case uses part (i) of the Free Variable Lemma:

> Abstraction $G \vdash A : B$ is $G \vdash (\lambda x : C.d) : (\Pi x : C.D)$, an immediate consequence of (1) $G, x : C \vdash d : D$ and (2) $G \vdash (\Pi x : C.D) : s$. By IH (on (1)) $FV(d), FV(D) \subseteq FV(G, x : C)$ and (on (2)) $FV(\Pi x : C.D), FV(s) \subseteq FV(G, x : C)$. Hence $FV(C) \subseteq FV(G)$ and $FV(\lambda x : C.d) \subseteq FV(G)$ since by definition $FV(\lambda x : C.d) = FV(C) \cup (FV(d) - \{x\})$ $(FV(\Pi x : C.D) = FV(C) \cup (FV(D) - \{x\}))$. Notice that by Free Variable Lemma (i) and (1), $x \notin FV(G)$. Therefore $FV(\lambda x : C.d), FV(\Pi x : C.D) \subseteq FV(G)$.

The *transfer*, *import* and *export*-cases again use the Free Variable definition $FV(G \boxdot \varepsilon) = FV(G)$.

Proof of (iii). The cases for *start* and *weakening* use the part (ii) of the Free Variable Lemma, we do the case for *start*:

> Start $G \vdash A : B$ is $G', x : C \vdash x : C$ a direct consequence of $G' \vdash C : s$. By IH for all A_i in G', $FV(A_i) \subseteq \{x_1, \ldots, x_{i-1}\}$. We have to prove that $FV(C) \subseteq \{x_1, \ldots, x_i\} = FV(G')$. But since $G' \vdash C : s$, we have $FV(C) \subseteq FV(G')$ by the previous clause (ii) of the Free Variable Lemma, hence $FV(A_i) \subseteq \{x_1, \ldots, x_{i-1}\}$ for all A_i in G.

For the *transfer*, *import* and *export*-cases we need the simple observation that A_i in G are the A_i in $G \boxdot \varepsilon$

3.2.3. LEMMA. **Start Lemma**
Let G be a legal generalized context. Then

(i) If $s_1 : s_2$ is a typing axiom ($\in \mathcal{A}^{Type}$), $G \vdash s_1 : s_2$

(ii) If $c : A$ is a logical axiom, ($\in \mathcal{A}^{Logic}$), $G \vdash c : A : Prop$

(iii) If $(x : A) \in \Gamma$ in $G' \boxdot \Gamma$ (where $G \equiv G' \boxdot \Gamma$), then $G' \boxdot \Gamma \vdash x : A$

PROOF. Proof of (i),(ii) and (iii). By assumption of $G \vdash A : B$ for some $A : B$. The result follows by induction on the derivation of $G \vdash A : B$.

Proof of (i). The cases for *transfer* and *import* all use the *transfer*$_1$-rule:

> Transfer$_2$ $G \vdash A : B$ is $G' \boxdot \varepsilon \vdash C : D$ where $G \equiv G' \boxdot \varepsilon$, an immediate consequence of $G' \vdash C : D : Type$. By IH $G' \vdash s_1 : s_2$, but then by transfer$_1$ $G' \boxdot \varepsilon \vdash s_1 : s_2$. Hence $G \vdash s_1 : s_2$, for $s_1 : s_2 \in \mathcal{A}^{Type}$.

For the *export*-cases we have look further back in the derivation in applying the IH, we show the case for K-export:

> K-export $G \vdash A : B$ is $G \vdash \hat{k}C : \Box D$ an immediate consequence of $G \boxdot \varepsilon \vdash C : D : Prop$. Applying the IH to the last step in the derivation does not work here, however since all derivations start from the context $\varepsilon; \varepsilon$ we can go up in the derivation tree to find the place where the $\boxdot$ was introduced going from G to $G \boxdot \varepsilon$ for the first time. This means something must have been derivable on G before, and since this derivation is shorter, IH gives that $G \vdash s_1 : s_2$, for $s_1 : s_2 \in \mathcal{A}^{Type}$.

The proof of (ii) is completely analogous.

Proof of (iii). The proof is straightforward for non-modal cases, and trivial for the *transfer* and *import*-rules:

> Transfer$_1$ $G \vdash A : B$ is $G' \boxdot \varepsilon \vdash C : s$ where $G \equiv G' \boxdot \varepsilon$, an immediate consequence of $G' \vdash C : s$. Note that this cannot occur when G is 'non-blocked' ($G \equiv \Gamma$). Therefore we treat the case of the 'complex' context $G' \boxdot \varepsilon$. Since $\Gamma \equiv \varepsilon$, it contains no variables, and so trivially $G \vdash x : C$ if $(x : C) \in \Gamma$.

For the *export*-cases we use an argument similar to the one given above in the proof of (i).

Given the following definitions of substitution on, and concatenation of generalized contexts, a Substition Lemma can be proved:

3.2.4. DEFINITION. **Subtitution, concatenation**

> On a generalized context $\Delta \equiv \Delta_1 \boxdot \ldots \boxdot \Delta_n$, the *substitution* of a term D for a variable x yields $\Delta[x := D] \equiv \Delta_1[x := D] \boxdot \ldots \boxdot \Delta_n[x := D]$.
>
> Given two generalized contexts $G \equiv \Gamma_1 \boxdot \ldots \boxdot \Gamma_m$ and $\Delta \equiv \Delta_1 \boxdot \ldots \boxdot \Delta_n$, their *concatenation* $G, \Delta \equiv \Gamma_1 \boxdot \ldots \boxdot \Gamma_m, \Delta_1 \boxdot \ldots \boxdot \Delta_n$.

3.2.5. LEMMA. **Substitution Lemma**
Assume (1) $G, x : C, \Delta \vdash A : B$ *and* (2) $G \vdash D : C$, *where* G *and* Δ *are generalized pseudo-contexts. Then* $G, \Delta[x := D] \vdash A[x := D] : B[x := D]$.

PROOF. Proof. By induction on (the length of the) derivation of (1), where M^* is used as an abbreviation for $M[x := D]$.

The non-modal cases are analogous to those in the proof of the Substitution Lemma for PTS's. The modal cases require some calculations with the definitions for substitution, e.g. :

> K-export $G, x : C, \Delta \vdash A : B$ is $G, x : C, \vdash \hat{k}E : \Box F$ an immediate consequence of $G, x : C, \Delta \boxdot \varepsilon \vdash E : F : Prop$. By IH $G, (\Delta \boxdot \varepsilon)^* \vdash E^* : F^*(: Prop)$. By definition $(\Delta \boxdot \varepsilon)^* \equiv \Delta^* \boxdot \varepsilon^* = \Delta^* \boxdot \varepsilon$, since $\varepsilon^* \equiv \varepsilon$. Hence $G, (\Delta \boxdot \varepsilon)^* \equiv G, \Delta^* \boxdot \varepsilon$, and so $G, \Delta^* \boxdot \varepsilon \vdash E^* : F^*(: Prop)$. Therefore by K-export $G, \Delta^* \vdash \hat{k}E^* : \Box(F^*)$, and since $(\hat{k}E^*) \equiv (\hat{k}E)^*$ $(FV(\hat{k}E) = FV(E))$ and $\Box(F^*) \equiv (\Box F)^*$, $G, \Delta^* \vdash (\hat{k}E)^* : (\Box F)^*$.

The proof of a Thinning Lemma is not completely straightforward. The *transfer* and *import*-rules are formulated in such a way that they yield a new generalized context of the form $G \boxdot \varepsilon$. However, to prove Thinning we have to show that there are derived versions of these rules that yield generalized contexts $G \boxdot \Gamma$, for an arbitrary 'non-blocked' context Γ.

To prove this, the following lemma is needed:

3.2.6. LEMMA. **Legality Lemma**
If $G \boxdot \Gamma', x : C$ is legal then $G \boxdot \Gamma' \vdash C : s$.

PROOF. By induction on the length of the derivation of $G \boxdot \Gamma', x : C \vdash A : B$. Except for the *axiom* cases which cannot occur (since $G \boxdot \Gamma', x : C \not\equiv \varepsilon$) and *start* and *weakening* which are immediate, the non-modal cases are regular. The *transfer* and *import*-cases cannot occur:

> Transfer$_1$ $G \boxdot \Gamma', x : C \vdash A : B$ is $G \boxdot \varepsilon \vdash D : s$, an immediate consequence of $G \vdash D : s$. This case cannot occur: $G \boxdot \Gamma', x : C \not\equiv G \boxdot \varepsilon$.

and the *export*-cases require some additional reasoning:

> K-export $G \boxdot \Gamma', x : C \vdash A : B$ is $G \vdash \hat{k}D : \Box E$ an immediate consequence of $G \boxdot \Gamma', x : C \boxdot \varepsilon \vdash D : E : Prop$. Since all derivations start from ε and are finite, we can go up in the tree to find the place where the $\boxdot$ was introduced, going from $G \boxdot \Gamma', x : C$, to $G \boxdot \Gamma', x : C \boxdot \varepsilon$ for the first time. This means that something must have been derivable on $G \boxdot \Gamma', x : C$ before, and since this derivation is shorter IH gives us that $G \boxdot \Gamma' \vdash C : s$.

3.2.7. LEMMA. **Derived Rules Lemma**
The following are derived rules in an MPTS:

$$Transfer'_1 \frac{G \vdash A : s}{G \boxdot \Gamma \vdash A : s} \qquad Transfer'_2 \frac{G \vdash A : B : Type}{G \boxdot \Gamma \vdash A : B}$$

$$Transfer'_3 \frac{G \vdash A : B : Set}{G \boxdot \Gamma \vdash A : B} \qquad Transfer'_{ax} \frac{G \vdash c : A : Prop}{G \boxdot \Gamma \vdash c : A}$$

$$K\,import' \frac{G \vdash A : \Box B : Prop}{G \boxdot \Gamma \vdash \check{k}A : B} \qquad 4\,import' \frac{G \vdash A : \Box B : Prop}{G \boxdot \Gamma \vdash \check{4}A : \Box B}$$

$$5\,import' \frac{G \vdash A : \neg\Box B : Prop}{G \boxdot \Gamma \vdash \check{5}A : \neg\Box B} \qquad B\,import' \frac{G \vdash A : B : Prop}{G \boxdot \Gamma \vdash \check{b}A : \neg\Box\neg B}$$

where Γ is a (non-blocked) pseudocontext such that $G \boxdot \Gamma$ is legal.

Given the original rules of Transfer and Import, proving the following is sufficient:

1 If $G \boxdot \varepsilon \vdash A : s$ then $G \boxdot \Gamma \vdash A : s$.

2 If $G \boxdot \varepsilon \vdash A : B(: \mathit{Type})$ then $G \boxdot \Gamma \vdash A : B$.

3 If $G \boxdot \varepsilon \vdash A : B(: \mathit{Set})$ then $G \boxdot \Gamma \vdash A : B$.

4 If $G \boxdot \varepsilon \vdash c : A(: \mathit{Prop})$ then $G \boxdot \Gamma \vdash c : A$.

5 If $G \boxdot \varepsilon \vdash \check{k}A : B$ then $G \boxdot \Gamma \vdash \check{k}A : B$

6 If $G \boxdot \varepsilon \vdash \check{4}A : \Box B$ then $G \boxdot \Gamma \vdash \check{4}A : \Box B$

7 If $G \boxdot \varepsilon \vdash \check{5}A : \neg\Box B$ then $G \boxdot \Gamma \vdash \check{5}A : \neg\Box B$

8 If $G \boxdot \varepsilon \vdash \check{b}A : \neg\Box\neg B$ then $G \boxdot \Gamma \vdash \check{b}A : \neg\Box\neg B$

PROOF. By induction on the length of Γ.
The basic case for $\Gamma \equiv \varepsilon$ is immediate by the above. The induction case where $\Gamma \equiv \Gamma', x : C$ is the same for all cases, we show 1:

1 By IH $G \boxdot \Gamma' \vdash A : s$, and by the Legality Lemma ($G \boxdot \Gamma'$ is legal) $G \boxdot \Gamma' \vdash C : s$, hence *weakening* yields $G \boxdot \Gamma', x : C \vdash A : s$ and $G \boxdot \Gamma \vdash A : s$.

Now we can prove a Thinning Lemma for the modal systems, using the 'subset relation' for generalized contexts defined earlier.

3.2.8. LEMMA. **Thinning Lemma**
Let G and Δ be legal generalized pseudocontexts such that $G \subseteq \Delta$. Then if $G \vdash A : B$, $\Delta \vdash A : B$.

PROOF. By induction on the length of the derivation of $G \vdash A : B$. The cases for *transfer* and *import* require the derived forms of the these rules from the derived rules lemma. We show the case for K-import:

> K-import $G \vdash A : B$ is $G' \boxdot \varepsilon \vdash \check{k}C : D$ where $G \equiv G' \boxdot \varepsilon$, an immediate consequence of $G' \vdash C : \Box D : \mathit{Prop}$. Since $G \equiv G' \boxdot \varepsilon$ and $G \subseteq \Delta$, it must be the case that $\Delta \equiv \Delta' \boxdot \Gamma$ for some Γ (since for all Γ, $\varepsilon \subseteq \Gamma$) and $G' \subseteq \Delta'$. Hence by IH $\Delta' \vdash C : \Box D : \mathit{Prop}$, so by the derived rule K-import' $\Delta' \boxdot \Gamma \vdash \check{k}C : D$ and therefore $\Delta \vdash \check{k}C : D$.

In the *Export*-cases we have to show that Δ is legal before the IH can be applied:

> K-export $G \vdash A : B$ is $G \vdash \hat{k}C : \Box D$ an immediate consequence of $G \boxdot \varepsilon \vdash C : D : \mathit{Prop}$. Since $G \subseteq \Delta$ and $\varepsilon \subseteq \varepsilon$, by definition $G \boxdot \varepsilon \subseteq \Delta \boxdot \varepsilon$. Furthermore $\Delta \boxdot \varepsilon$ is legal: Δ is legal, hence by the Start Lemma (i) $\Delta \vdash s_1 : s_2$ for $s_1 : s_2 \in \mathcal{A}^{\mathit{Type}}$ (note that $\mathcal{A}^{\mathit{Type}} \neq \emptyset$), and by *transfer*$_1$, $\Delta \boxdot \varepsilon \vdash s_1 : s_2$. Therefore by IH $\Delta \boxdot \varepsilon \vdash C : D : \mathit{Prop}$, and so $\Delta \vdash \hat{k}C : \Box D$.

3.2.9. COROLLARY. **Strong Thinning**
For terms (A) that are not proofs (not $A : B : Prop$), we can prove a stronger result (for cases where G is an intial part of Δ) by combining Thinning with the transfer rule.

Let G and Δ be legal generalized pseudocontexts such that $G \leq \Delta$. Then

(i) if $G \vdash A : s$, $\Delta \vdash A : s$.

(ii) if $G \vdash A : B : Type$, $\Delta \vdash A : B : Type$.

(iii) if $G \vdash A : B : Set$, $\Delta \vdash A : B : Set$.

PROOF. Proof of (i), (ii), and (iii). By construction of Δ from G while preserving the derivability of $A : s$. We do the proof of (ii):
Suppose that $G \vdash A : B : Type$ for some terms A and B:

(1) $\Gamma_1 \boxdot \ldots \boxdot \Gamma_m \vdash A : B : Type$

Since $\forall i(1 \leq i \leq m)(\Gamma_i \subseteq \Gamma'_i)$, we can conclude by Thinning

(2) $\Gamma'_1 \boxdot \ldots \boxdot \Gamma'_m \vdash A : B : Type$

Now the transfer$_2$ rule can be applied to obtain

$$\Gamma'_1 \boxdot \ldots \boxdot \Gamma'_m \boxdot \varepsilon \vdash A : B$$

and by transfer$_1$ we have

$$\Gamma'_1 \boxdot \ldots \boxdot \Gamma'_m \boxdot \varepsilon \vdash B : Type$$

hence

(3) $\Gamma'_1 \boxdot \ldots \boxdot \Gamma'_m \boxdot \varepsilon \vdash A : B : Type$

Since $\varepsilon \subseteq \Gamma'_{m+1}$, Thinning can yield

(4) $\Gamma'_1 \boxdot \ldots \boxdot \Gamma'_m \boxdot \Gamma'_{m+1} \vdash A : s$

By repeating these last two steps, transfer$_2$ and transfer$_1$ to introduce a new $\boxdot$ and Thinning to expand ε into the respective Γ'_j $(m+1 \leq j \leq n)$, we obtain

(5) $\Gamma'_1 \boxdot \ldots \boxdot \Gamma'_m \boxdot \ldots \boxdot \Gamma'_n \vdash A : B : Type$

And thereby we have shown that $A : B : Type$ is derivable on the generalized context Δ:

(6) $\Delta \vdash A : B : Type$

3.3. The Stripping Lemma

Before we can state and prove a sufficiently strong Stripping Lemma for the MPTS's, we have to show that for 'non-proof-terms' (terms A for which there is no context G and a term B such that $G \vdash A : B : Prop$) it is 'business as usual'. That is to say that if such a term can be constructed in a context, it can also be constructed in a subcontext of that context and vice versa. We do this by showing that in derivations of non proof terms, $\square$s can be inserted in and deleted from the context.

Proving Insertion and Deletion requires that a subterm of a term that is not a proof term is itself not a proof term (Subterm Lemma). This is a property of the PTSs in the logic cube, and we can prove that it holds for the logical MPTSs by means of a simple mapping of MPTS-terms on PTS-terms. Here is a sketch of the proof route:

Definition of the proof terms and non proofs terms in PTSs

Proof of the Subterm Lemma for PTSs

Proof of the Subterm Lemma for MPTSs using the mapping '$|\ |$'

Proof of the Insertion Lemma

Proof of the Deletion Lemma

3.3.1. DEFINITION. Following [Barendregt 1992], the terms of the *PTSs* in the Logic Cube can be divided into:

i A is a *set-kind*: $\exists\Gamma[\Gamma \vdash A : Type^s]$

ii A is a *prop-kind*: $\exists\Gamma[\Gamma \vdash A : Type^p]$

iii A is a *set-constructor*: $\exists\Gamma, B[\Gamma \vdash A : B : Type^s]$

iv A is a *prop-constructor*: $\exists\Gamma, B[\Gamma \vdash A : B : Type^p]$

v A is a *set*: $\exists\Gamma[\Gamma \vdash A : Set]$

vi A is a *proposition*: $\exists\Gamma[\Gamma \vdash A : Prop]$

vii A is an *element*: $\exists\Gamma, B[\Gamma \vdash A : B : Set]$

viii A is a *proof term*: $\exists\Gamma, B[\Gamma \vdash A : B : Prop]$

Hence A is *not a proof term* in cases *i* - *vii*, and A is *a proof term* in case *viii*.

In the following we need that the sets of non-proof terms (*i* - *vii*) are disjunct from the set of proof terms (*viii*). This can be established using some results from [Geuvers 1993]. In his formulation of PTS_{β}s, there are no constants ($c \in C$) inhabiting the logical axioms, but the following results and their proofs can easily be seen to go through for our formulation (the constants behave as a subset of the *Prop*-variables, Var^{Prop}).

3.3.2. DEFINITION. **(Geuvers) s-Term, s-Elt**

For Γ a context, s a sort and A a term, A *is an s-term in* Γ (notation $A \in s$-Term(Γ)) if $\Gamma \vdash A : s$,

For Γ a context, s a sort and A a term, A *is an s-element in* Γ (notation $A \in s$-Elt(Γ)) if $\Gamma \vdash A : B : s$ for some term B.

Given these notions, Geuvers goes on to show that for injective PTS$_\beta$s, with sorted variables, the following holds.

3.3.3. LEMMA. **(Geuvers) Classification Lemma for injective systems**
For s, s' sorts and $s \not\equiv s'$

1 $s\text{-}Term \cap s'\text{-}Term = \emptyset$

2 $s\text{-}Elt \cap s'\text{-}Elt = \emptyset$

By 1, we immediately have that the sets (i), (ii), (v), and (vi) are all disjoint. Likewise, by 2, (iii), (iv), (vii), and $(viii)$ are all disjoint. Therefore what remains to be shown is that the set of proof terms, $(viii)$, is disjoint from (i), (ii), (v), and (vi). To do this, we require a further notion.

3.3.4. DEFINITION. **(Geuvers) Heart of a pseudoterm**
The *heart of a pseudoterm* A, $h(A)$, is defined by induction on the structure of terms as follows.

$$\begin{array}{rcl} h(s) & := & s, \text{ for } s \in \mathcal{S}, \\ h(x) & := & x, \text{ for } x \in Var, \\ h(\Pi x : B.C) & := & h(C), \\ h(\lambda x : B.M) & := & h(M), \\ h(MN) & := & h(M). \end{array}$$

For the hearts of s-Terms and s-Elts, the following holds.

3.3.5. LEMMA. **(Geuvers)**
For an injective PTS_β with all rules of the form (s_1, s_2) we have

$$\begin{array}{rcl} M \in s\text{-}Elt & \Leftrightarrow & h(M) = x \in Var^s \vee \\ & & h(M) = s'' \text{ with } s'' : s' \text{ and } s' : s \in \mathcal{A}^{Type} \text{ for some } s' \in \mathcal{S} \\ M \in s\text{-}Term & \Rightarrow & h(M) = x \in Var^{s'} \text{ with } s : s' \in \mathcal{A}^{Type} \vee \\ & & h(M) = s' \text{ with } s' : s \in \mathcal{A}^{Type}. \end{array}$$

3.3.6. PROPOSITION. *For the $PTS_\beta s$ in the Logic Cube, the set of proof terms is disjoint from the sets of non-proof terms.*

PROOF. By the Classification Lemma we have that the set of proof terms, (*viii*), is disjoint from the non-proof termsets (*iii*), (*iv*), and (*vii*). Hence what remains to be shown is that (*viii*) is disjoint from (*i*), (*ii*), (*v*), and (*vi*).

Suppose that this is not true (towards a contradiction). Then there exists a proof term $M(: A : Prop)$, that is both an s-Elt and an s-Term. Hence by the last lemma and the fact that M is an s-Elt we have (1) $h(M) = x \in Var^{Prop}$. By the fact that M is an s-Term, either (2) $h(M) \in Var^{s'}$ with $s' \in \mathcal{A}^{Type}$, or (3) $h(M) = s'$ where $s' : s \in \mathcal{A}^{Type}$. Since by (1) $x \in Var^{Prop}$, it cannot be the case that $h(M) = s'$ for any $s' \in \mathcal{S}$. It is also impossible that $h(M) = x \in Var^{s'}$ with $s : s' \in \mathcal{A}^{Type}$, since $\mathcal{A}^{Type}$ does not contain a typing axiom of the form $s : Prop$. Both (2) and (3) are in contradiction with (1), hence there cannot exist a proof term that is an s-Elt as well as an s-Term. Therefore (*viii*) is disjoint from (*i*), (*ii*), (*v*), and (*vi*).

For PTSs in the Logic Cube the definition of subexpressions is an extension of the usual definition for PTSs: M *sub* A iff $N \in Sub(A)$, where $Sub(A)$, the set of subexpressions of A, is defined as follows.

$Sub(A) = \{A\}$, if A is one of the constants (sorts) or variables

$Sub(A) = \{A\} \cup Sub(P) \cup Sub(Q)$, if A is of the form $\Pi x : P.Q, \lambda x : P.Q$ or PQ

3.3.7. LEMMA. **Subterm Lemma for the Logic Cube**
If A is a PTS-term and A is not a proof term, then if B is subterm of A, B is not a proof term.

PROOF. By induction on the structure of A.

$A \equiv s(s \in \mathcal{S})$ and A is not a proof term. Since $Sub(A) = Sub(s) = \{s\} = \{A\}$, we are done.

$A \equiv x(x \in Var)$ and A is not a proof, that is $x \in Var^s$ for $s = Set, Type^s, Type^p$. Since $Sub(A) = Sub(x) = \{x\} = \{A\}$, we are done.

$A \equiv \Pi x : C.D$ and A is not a proof. Since A is not a proof term it is typeable, and hence by the Stripping Lemma there exists a context Γ such that $\Gamma \vdash (\Pi x : C.D) : s_3$ and (1) $\Gamma \vdash C : s_1$ and (2) $\Gamma, x : C \vdash D : s_2$ for some $s_1, s_2, s \in \mathcal{S}$ and, $s \equiv s_3, (s_1, s_2, s_3) \in \mathcal{R}$. But then from (1) : $\exists \Gamma'[\Gamma' \vdash C : s_1]$ $(\Gamma' \equiv \Gamma)$ and $s_1 \in \mathcal{S}$, hence by definition (using the partitioning of the terms) C is not a proof. And from (2) : $\exists \Gamma'[\Gamma' \vdash D : s_2]$ $(\Gamma' \equiv \Gamma, x : C)$ and $s_2 \in \mathcal{S}$, hence by definition (using the partitioning of the terms) D is not a proof.

$A \equiv \lambda x : M.N$ and A is not a proof. Since A is not a proof term it is typeable, and hence by the Stripping Lemma there exists a context Γ such that $\Gamma \vdash (\lambda x : M.N) : (\Pi x : M.C)$, and
(1) $\Gamma \vdash (\Pi x : M.C) : s_3$ and (2) $\Gamma, x : M \vdash N : C$. From (1) we obtain by the Stripping Lemma that (3) $\Gamma \vdash M : s_1$ and (4) $\Gamma, x : M \vdash C : s_2$. Since we know that $(\lambda x : M.N)$ is not a proof, $s_3 \neq Prop$. But then, by inspection of the rules in $\mathcal{R}$, $s_1 \neq Prop$ and $s_2 \neq Prop$, hence by (3) M is not a proof term and by (2) and (4) N is not a proof term.

$A \equiv MN$ and A is not a proof. Since A is not a proof term it is typeable, and hence by the Stripping Lemma there exists a context Γ such that (1) $\Gamma \vdash MN : D[x := N]$ and (2) $\Gamma \vdash M : (\Pi x : C.D)$ and (3) $\Gamma \vdash N : C$. From Stripping on (2) we have (4) $\Gamma \vdash C : s_1$, (5) $\Gamma, x : C \vdash D : s_2$ and (6) $\Gamma \vdash (\Pi x : C.D) : s_3$ for $s_1, s_2, s_3 \in \mathcal{S}$ and $(s_1, s_2, s_3) \in \mathcal{R}$. But then we have by the Substitution Lemma on (3) and (5) that (7) $\Gamma \vdash D[x := N] : s_2$. Since we know that MN is not a proof term, $s_2 \neq Prop$. Inspection of the rules in $\mathcal{R}$ shows that for all rules where $s_2 \neq Prop$, $s_1 \neq Prop$, and $s_3 \neq Prop$. Hence combining, (3) and (4), N is not a proof term and, by (2) and (6), M is not a proof term.

Note that the case that $M \equiv c$ for some logical axiom constant c could not occur, since by definitions given above $c : A : Prop$ is always a proof term.

3.3.8. DEFINITION. **Erasure Mapping**
Let $|\;|$ be a mapping of MPTS-terms to PTS-terms, which 'erases' the modal terms:

i $|\Box A| = |A|$, $|\neg\Box\neg A| = |A|$

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 $|\Gamma, x : A| = |\Gamma|, |x : A|$, $|G \boxdot \Gamma| = |G|, |\Gamma|$

iv $|A : B| = |A| : |B|$, $|\varepsilon| = \varepsilon$, $|x| = x$ (for $x \in Var$), $|s| = s$ (for $s \in \mathcal{S}$)

v $|\check{k}A| = |A|$, $|\check{b}A| = |A|$, $|\check{4}A| = |A|$, $|\check{5}A| = |A|$,
$|\hat{k}A| = |A|$, $|\hat{d}A| = |A|$, $|\hat{t}A| = |A|$.

This mapping reduces MPTSs to PTSs by simply erasing everything that is 'modal' about them. It preserves typeability and β-reductions in MPTS-terms, which shows that 'erasing' a derivation in a certain MPTS in the Modal Logic Cube, will yield a correct derivation in the corresponding PTS in the Logic Cube.

3.3.9. LEMMA. **Preservation of Substitution**
$|\;|$ preserves substitutions:
for MPTS-terms A and B $|A|[x := |B|] = |A[x := B]|$.

PROOF. By induction on the structure of the term A.

3.3.10. LEMMA. **Preservation of β-reduction**
$|\;|$ preserves β-reductions: for MPTS-terms A, M and N

1a *If $(\lambda x : A.M)N \to_\beta M[x := N]$ then $|(\lambda x : A.M)N| \to_\beta |M[x := N]|$.*

1b *If $|M| \to_\beta |N|$, then*

1 $|ZN| \to_\beta |ZM|$
2 $|NZ| \to_\beta |MZ|$
3 $|\lambda x.N| \to_\beta |\lambda x.M|$

PROOF. 1a $|(\lambda x : A.M)N| = |(\lambda x : A.M)||N| = (\lambda x : |A|.|M|)|N| \to_\beta |M|[x := |N|]$, by the Substitution Preservation Lemma $|M|[x := |N|] = |M[x := N]|$.

1b If $|M| \to_\beta |N|$, then

1 $|ZM| =_{ii} |Z||M| \to_\beta |Z||N| =_{ii} |ZN|$

2 $|MZ| =_{ii} |M||Z| \to_\beta |N||Z| =_{ii} |NZ|$

3 $|\lambda x : A.M| =_{ii} \lambda x : |A|.|M| \to_\beta \lambda x : |A|.|N| =_{ii} |\lambda x : A.N|$

3.3.11. LEMMA. **Preservation of Typeability**
| | preserves typeability: $G \vdash A : B \Rightarrow |G| \vdash |A| : |B|$.

PROOF. By induction on the derivation of $G \vdash A : B$

Using the erasure map we can distinguish between proof-terms and non-proofterms in MPTSs.

3.3.12. DEFINITION. The terms of the MPTSs in the Logic Cube can be divided into the same subsets as those of the PTSs, by mapping them to PTS-terms:

i A is a *set-kind*: $\exists\Gamma[\Gamma \vdash |A| : Type^s]$

ii A is a *prop-kind*: $\exists\Gamma[\Gamma \vdash |A| : Type^p]$

iii A is a *set-constructor*: $\exists\Gamma, B[\Gamma \vdash |A| : B : Type^s]$

iv A is a *prop-constructor*: $\exists\Gamma, B[\Gamma \vdash |A| : B : Type^p]$

v A is a *set*: $\exists\Gamma[\Gamma \vdash |A| : Set]$

vi A is a *proposition*: $\exists\Gamma[\Gamma \vdash |A| : Prop]$

vii A is an *element*: $\exists\Gamma, B[\Gamma \vdash |A| : B : Set]$

viii A is a *proof term*: $\exists\Gamma, B[\Gamma \vdash |A| : B : Prop]$

Hence A is *not a proof term* in cases *i* - *vii*, and A is *a proof term* in case *viii*.

3.3.13. DEFINITION. **Subexpressions**
For MPTS the definition of subexpressions is an extension of the definition for PTSs given above with the clause

$$Sub(A) = \{A\} \cup Sub(P), \text{ if } A \text{ is of the form } \Box P, \breve{k}P, \breve{4}P, \breve{5}P, \breve{b}P, \text{ or } \hat{k}P, \hat{d}P, \hat{t}P.$$

'$\Box$','$\breve{k}$', '$\hat{k}$', and the like are not terms, and hence they cannot be subterms.

3.3.14. LEMMA. *If B is a subterm of A (A and B MPTS terms) then $|B|$ is a subterm of $|A|$.*

PROOF. By induction on the structure of the term A.

Given the Subterm Lemma for PTSs we can easily prove it for MPTSs

3.3.15. LEMMA. Subterm Lemma for the Modal Logic Cube
If A is a MPTS-term and A is not a proof term, then if B is subterm of A, B is not a proof term.

PROOF. (towards a contradiction). Suppose that

(1) A is not a proof: $\neg\exists\Gamma, C\ [\Gamma\ \vdash\ |A| : C : Prop]$

(2) B is a subterm of A

(3) B is a proof: $\exists\Gamma, C\ [\Gamma\ \vdash\ |A| : C : Prop]$

Then since B is a subterm of A, $|B|$ is a subterm of $|A|$ (Lemma). But then by the Subterm Lemma for PTS-terms $|B|$ is not a proof: $\neg\exists\Gamma, C\ [\Gamma\ \vdash\ |A| : C : Prop]$. This contradicts (3), hence if A is not a proof term and B is a subterm of A, B is not a proof.

With the help of the Subterm Lemma we can show that for the derivability of non-proof terms, the 'blocks' in the generalized contexts are of no interest: they can be removed or inserted at will.

3.3.16. LEMMA. Block Insertion Lemma
If G and G' are generalized contexts (for $n \geq 0$) and $G \equiv \Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)}, \Gamma_n$, $G' \equiv \Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)} \boxdot \Gamma_n$, then: if $G\ \vdash\ A : B$ and A is not a proof, then $G'\ \vdash\ A : B$.

PROOF. By induction on the derivation of $G\ \vdash\ A : B$. The non-modal cases are easy (use *transfer* in the *axiom* cases), with the exception of the *application* and *abstraction* case. These depend completely on the Subterm Lemma, we do the *abstraction*-case:

> Abstraction $G\ \vdash\ A : B$ is $\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)}, \Gamma_n\ \vdash\ (\lambda x : C.d) : (\Pi x : C.D)$, an immediate consequence of (1) $\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)}, \Gamma_n, x : C\ \vdash\ d : D$, and (2) $\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)}, \Gamma_n\ \vdash\ (\Pi x : C.D) : s$. Since (by hypothesis) $(\lambda x : C.d)$ is not a proof term, we have by the Subterm Lemma that d is not a proof term. Hence by IH on (1), $\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)} \boxdot \Gamma_n, x : C\ \vdash\ d : D$. By definition $(\Pi x : C.D)$ is not a proof term $(G\ \vdash\ (\Pi x : C.D) : s)$, and so by IH on (2)
> $\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)} \boxdot \Gamma_n\ \vdash\ (\Pi x : C.D) : s$. Therefore
> $\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)} \boxdot \Gamma_n\ \vdash\ (\lambda x : C.d) : (\Pi x : C.D)$, and so
> $G'\ \vdash\ (\lambda x : C.d) : (\Pi x : C.D)$.

Note that in the cases for the *transfer*-rules, we insert a block into an empty context ε, which subsequently splits into two empty contexts separated by a block:

> Transfer$_1$ $G\ \vdash\ A : B$ is $\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-2)} \boxdot \varepsilon\ \vdash\ C : s$ (where $\Gamma_{n-1} \equiv \Gamma_n \equiv \varepsilon$), an immediate consequence of $\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-2)}\ \vdash\ C : s$. But then by consecutive applications of transfer$_1$, $\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-2)} \boxdot \varepsilon\ \vdash\ C : s$, and $\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-2)} \boxdot \varepsilon \boxdot \varepsilon\ \vdash\ C : s$. Hence $\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-2)} \boxdot \Gamma_{n-1} \boxdot \Gamma_n\ \vdash\ C : s$, and so $G'\ \vdash\ C : s$.

The induction cases for the *import*- and *export*-rules are trivial, these cases cannot occur since *import*- and *export*-terms are proof terms. However, we have to show this using the erasure mapping:

K-import $G \vdash A : B$ is $\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)} \boxdot \varepsilon \vdash \check{k}C : D$ (where $\Gamma_n \equiv \varepsilon$), an immediate consequence of (1) $\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)} \vdash C : \Box D : Prop$, and $\check{k}C$ is not a proof term. This case cannot occur, $\check{k}C$ is a proof term: from $\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)} \boxdot \varepsilon \vdash \check{k}C : D$ by Preservation of Typability (2) $|\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)} \boxdot \varepsilon| \vdash |\check{k}C| : |D|$. Since $|\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)} \boxdot \varepsilon| =_{iii} |\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)}, \varepsilon| = |\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)}|$, we have (3) $|\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)}| \vdash |\check{k}C| : |D|$. Furthermore, from (1) $\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)} \vdash \Box D : Prop$, and hence
(4) $|\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)}| \vdash |\Box D| : |Prop|$. Since $|\Box D| =_i |D|$ and $|Prop| =_{iv} Prop$,
(5) $|\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)}| \vdash |D| : Prop$, which combined with (4) yields
$|\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)}| \vdash |\check{k}C| : |D| : Prop$. Hence $\exists \Gamma, B[\Gamma \vdash |\check{k}C| : B : Prop]$
($\Gamma = |\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)}| =_{iii} |\Gamma_1, \ldots, \Gamma_{(n-1)}|$ and $B = |D|$), and so $\check{k}C$ is a proof term.

3.3.17. LEMMA. **Block Deletion Lemma**
If G and G' are generalized contexts (for $n \geq 2$) and $G \equiv \Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)} \boxdot \Gamma_n$, $G' \equiv \Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)}, \Gamma_n$, then: if $G \vdash A : B$ and A is not a proof term, then $G' \vdash A : B$.

PROOF. By induction on the derivation of $G \vdash A : B$.
The proof is similar to that of the Block Insertion Lemma. The Subterm Lemma is again needed for the *application* and the *abstraction*-case, this time we do the *application* case:

Application $G \vdash A : B$ is $\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)} \boxdot \Gamma_n \vdash FN : D[x := N]$, an immediate consequence of (1) $\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)} \boxdot \Gamma_n \vdash F : (\Pi x : C.D)$, and (2) $\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)} \boxdot \Gamma_n \vdash N : C$. Since (by hypothesis) FN is not a proof term, we have by the Subterm Lemma that F is not a proof term and N is not a proof term. Hence by IH on (1) and (2),
$\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)}, \Gamma_n \vdash F : (\Pi x : C.D)$, and $\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)}, \Gamma_n \vdash N : C$.
Therefore $\Gamma_1 \boxdot \ldots \boxdot \Gamma_{(n-1)}, \Gamma_n \vdash FN : D[x := N]$, and so $G' \vdash FN : D[x := N]$.

For the *transfer*-cases we need that $\Gamma, \varepsilon = \Gamma$. The *axiom*-cases cannot occur since by the assumptions of the Lemma G contains at least one block and hence cannot be ε. Showing that the *import* and *export*-cases cannot occur, is again done by means of the erasure map.

3.3.18. COROLLARY. **Refsnart**
Given the lemma it is easy to see that the following 'inversions' of the transfer rules are derived rules in the MPTSs:

$$\text{refsnart}_1 \frac{G \boxdot \varepsilon \vdash A : s}{G \vdash A : s}$$

$$\text{refsnart}_2 \frac{G \boxdot \varepsilon \vdash A : B : Type}{G \vdash A : B}$$

$$\text{refsnart}_3 \frac{G \boxdot \varepsilon \vdash A : B : Set}{G \vdash A : B}$$

PROOF.

Refsnart$_1$ If $G \boxdot \varepsilon \vdash A : s$, then by Block Deletion $G, \varepsilon \vdash A : s$, hence $G \vdash A : s$.

Refsnart$_2$ If $G \boxdot \varepsilon \vdash A : B : Type$, then by Block Deletion $G, \varepsilon \vdash A : B$, hence $G \vdash A : B$.

Refsnart$_3$ If $G \boxdot \varepsilon \vdash A : B : Set$, then by Block Deletion $G, \varepsilon \vdash A : B$, hence $G \vdash A : B$.

The Block Insertion and Deletion Lemmas show that non-proof terms can really 'travel freely' through generalized contexts. For these terms we can disregard whether they are derivable on a prefix or a subcontext of the generalized context in which we are carrying out our derivation, since they can always be made available in that context. This allows for a Stripping Lemma that is comparable in strength to that for $MPTS_\beta$.

3.3.19. LEMMA. **Stripping Lemma**
For G and G' generalized pseudocontexts, Γ a 'non-blocked' pseudocontext, and M, N and R terms, we have the following:

(i) $G \vdash s : R,\ (s \in \mathcal{S}) \Rightarrow R = s'$ *with* $s : s' \in \mathcal{A}^{Type}$ *for some* $s' \in \mathcal{S}$.
$G \vdash c : R,\ (c \in \mathcal{C}) \Rightarrow R = A$ *for some term* A *with* $c : A \in \mathcal{A}^{Logic}$.

(ii) $G \vdash x : R,\ (x \in Var) \Rightarrow R = A$ *with* $(x : A) \in G$ *for some term* A.

(iii) $G \vdash \Pi x : M.N : R \Rightarrow G \vdash M : s_1,\ G, x : M \vdash N : s_2$ *and* $R = s_3$ *with* $(s_1, s_2, s_3) \in \mathcal{R}$ *for some* $s_1, s_2, s_3 \in \mathcal{S}$.

(iv) $G \vdash \lambda x : M.N : R \Rightarrow G \vdash \Pi x : M.B : s,\ G, x : M \vdash N : B$ *and* $R = \Pi x : M.B$ *for some term* B *and* $s \in \mathcal{S}$.

(v) $G \vdash MN : R \Rightarrow G \vdash M : (\Pi x : A.B),\ G \vdash N : A$ *with* $R = B[x := N]$ *for some terms* A *and* B.

(vi) $G \vdash \Box N : R \Rightarrow G \vdash N : Prop$ *and* $R = Prop$.

(vii) $G \vdash \breve{k}M : R \Rightarrow G \equiv G' \boxdot \Gamma$, *and* $G' \vdash M : \Box A : Prop$ *for some term* A, *and* $R = A$.

(ix) $G \vdash \breve{4}M : R \Rightarrow G \equiv G' \boxdot \Gamma$, *and* $G' \vdash M : \Box A : Prop$ *for some term* A, *and* $R = \Box A$.

(x) $G \vdash \breve{5}M : R \Rightarrow G \equiv G' \boxdot \Gamma$, *and* $G' \vdash M : \neg\Box A : Prop$ *for some term* A, *and* $R = \neg\Box A$.

(xi) $G \vdash \breve{b}M : R \Rightarrow G \equiv G' \boxdot \Gamma$, *and* $G' \vdash M : A : Prop$ *for some term* A, *and* $R = \neg\Box\neg A$.

(xii) $G \vdash \hat{k}M : R \Rightarrow G \boxdot \varepsilon \vdash M : A : Prop$ *for some term* A, *and* $R = \Box A$.

(xiii) $G \vdash \hat{d}M : R \Rightarrow G \boxdot \varepsilon \vdash M : A : Prop$ *for some ter* A *and* $R = \neg\Box\neg A$.

(xiv) $G \vdash \hat{t}M : R \Rightarrow G \boxdot \varepsilon \vdash M : A : Prop$ *for some term* A, *and* $R = A$.

PROOF. By induction on the derivation of $G \vdash P : R$. The cases of $G \vdash P : R$ are distinguished according to the form of P.

The proof is mainly a lot of work, the only cases where the difficulties with *transfer* really come into play are those for *abstraction* and *application*, we do the *application* case:

$P \equiv MN$ $G \vdash MN : R$. Then P could only have been derived using (1) *application*, (2) *weakening*, (3) *conversion*, (4) *transfer* as final rule.

(1) In this case $G \vdash M : (\Pi x : A.B)$. $G \vdash N : A$ with $R \equiv B[x := N]$ for some terms A and B.
Since '$\equiv$' implies '$=$', $R = B[x := N]$, and so $G \vdash M : (\Pi x : A.B)$, $G \vdash N : A$ with $R = B[x := N]$ for some terms A and B.

(2) In this case $G \equiv G_1, y : C$ for some term C where $G_1 \vdash C : s$ for some $s \in \mathcal{S}$, and $G_1 \vdash MN : R$. By IH $G_1 \vdash M : (\Pi x : A.B)$, $G_1 \vdash N : A$ with $R = B[x := N]$ for some terms A and B. In this case we have by Thinning that $G \vdash M : (\Pi x : A.B)$, respectively $G \vdash N : A$ with $R = B[x := N]$ for some terms A and B.

(3) In this case $G \vdash MN : R'$ where $R = R'$. By IH, $G \vdash M : (\Pi x : A.B)$, $G \vdash N : A$ with $R' = B[x := N]$ for some terms A and B. Since $R = R'$, $R = B[x := N]$, and so $G \vdash M : (\Pi x : A.B)$, $G \vdash N : A$ with $R = B[x := N]$ for some terms A and B.

(4) In this case the last rule was *transfer*$_1$, *transfer*$_2$, or *transfer*$_3$

Transfer$_1$ In this case $G \equiv G' \boxdot \varepsilon$, and $G' \vdash MN : R$ and $R \equiv s$ for some $s \in \mathcal{S}$. By IH $G' \vdash M : (\Pi x : A.B)$, $G' \vdash N : A$, and $R = B[x := N](= s)$ for some terms A and B. By definition MN is 'not a proof': $G' \vdash MN : s$, hence M and N are not proofs (Subterm Lemma) and therefore we have $G' \boxdot \varepsilon \vdash M : (\Pi x : A.B)$ and $G' \boxdot \varepsilon \vdash N : A$ from $G', \varepsilon \vdash M : (\Pi x : A.B)$ and $G', \varepsilon \vdash N : A$ by Block Insertion. Therefore $G \vdash M : (\Pi x : A.B)$, $G \vdash N : A$ with $R = B[x := N]$ for some terms A and B.

Transfer$_2$ In this case $G \equiv G' \boxdot \varepsilon$, and $G' \vdash MN : R : \mathit{Type}$. By IH $G' \vdash M : (\Pi x : A.B)$, $G' \vdash N : A$, and $R = B[x := N]$ for some terms A and B. By definition MN is 'not a proof': $G' \vdash MN : R : \mathit{Type}$, hence M and N are not proofs (Subterm Lemma) and therefore we have $G' \boxdot \varepsilon \vdash M : (\Pi x : A.B)$ and $G' \boxdot \varepsilon \vdash N : A$ from $G', \varepsilon \vdash M : (\Pi x : A.B)$ and $G', \varepsilon \vdash N : A$ by Block Insertion. Therefore $G \vdash M : (\Pi x : A.B)$, $G \vdash N : A$ with $R = B[x := N]$ for some terms A and B.

Transfer$_3$ In this case $G \equiv G' \boxdot \varepsilon$, and $G' \vdash MN : R : \mathit{Set}$. By IH $G' \vdash M : (\Pi x : A.B)$, $G' \vdash N : A$, and $R = B[x := N]$ for some terms A and B. By definition MN is 'not a proof': $G' \vdash MN : R : \mathit{Set}$, hence M and N are not proofs (Subterm Lemma) and therefore we have $G' \boxdot \varepsilon \vdash M : (\Pi x : A.B)$ and $G' \boxdot \varepsilon \vdash N : A$ from $G', \varepsilon \vdash M : (\Pi x : A.B)$ and $G', \varepsilon \vdash N : A$ by Block Insertion. Therefore $G \vdash M : (\Pi x : A.B)$, $G \vdash N : A$ with $R = B[x := N]$ for some terms A and B.

Hence in all cases: $G \vdash M : (\Pi x : A.B)$, $G \vdash N : A$ with $R = B[x := N]$ for some terms A and B.

We show one modal case, to give an idea how the *import* and *export*-rules interact with *weakening* and *conversion*.

$P \equiv \check{k}M$ $G \vdash \check{k}M : R$. Then P could only have been derived using (1) *import*, (2) *weakening*, (3) *conversion*, or (4) *transfer* as final rule.

(1) In this case $G \equiv G' \boxdot \varepsilon$, and $G' \vdash M : \Box A$ for some term A, and $R \equiv A$. Since '$\equiv$' implies '$=$' and ε is a non-blocked pseudo context, $R = A$ and $G \equiv G' \boxdot \Gamma$. Hence $G \equiv G' \boxdot \Gamma$ and $G' \vdash M : A$ for some term A, and $R = A$.

(2) In this case $G \equiv G_1, y : C$ for some term C where $G_1 \vdash C : s$ for some $s \in \mathcal{S}$, and $G_1 \vdash \check{k}M : R$. By IH $G_1 \equiv G' \boxdot \Gamma'$, and $G' \vdash M : \Box A$ for some term A, In this case, $G \equiv G' \boxdot \Gamma', x : C$ and since $\Gamma', x : C$ is again a non-blocked pseudo context (Γ), $G \equiv G' \boxdot \Gamma$ and $G' \vdash M : \Box A$ for some term A. Therefore $G \equiv G' \boxdot \Gamma$ and $G' \vdash M : A$ for some term A, and $R = A$.

(3) In this case $G \vdash \check{k}M : R'$ where $R = R'$. By IH, $G \equiv G' \boxdot \Gamma$, and $G' \vdash M : \Box A$ for some term A, and $R' = A$. Since $R = R'$, $R = A$, and so $G \equiv G' \boxdot \Gamma$ and $G' \vdash M : A$ for some term A, and $R = A$.

(4) In this case $G \equiv G' \boxdot \varepsilon$ and $G' \vdash \check{k}C : R$. This case cannot occur, $\check{k}C$ is by definition a 'proof', and there is no *transfer* rule that accepts a proof term as its premise.

Hence in all cases: $G \equiv G' \boxdot \Gamma$ and $G' \vdash M : A$ for some term A, and $R = A$.

Note that this formulation of the Stripping Lemma treats variables in a uniform way. However, keeping in mind the distinction between proofs and non proofs we could have been more concise in the case of the proof variables ($x : A$ where A is a proposition): proof variables cannot be transported over a '$\boxdot$' by the Transfer rules, and are hence elements of the final non-blocked context of G.

(ii') $G \vdash x : R$, ($x \in Var^{Prop}$) $\Rightarrow$ $R = A$ with $x : A \in G$ for some term A (where $G \equiv \Gamma$ or $G \equiv G' \boxdot \Gamma$).

PROOF. $P \equiv x(x \in Var^{Prop})$ $G \vdash x : R$, ($x \in Var^{Prop}$). Then P could only have been derived using (1) *start*, (2) *weakening*, (3) *conversion*, or (4) *transfer* as final rule.

(1) In this case $R \equiv A$ with $x : A \in \Gamma$ for some term A (where $G \equiv \Gamma$ or $G \equiv G' \boxdot \Gamma$). Since '$\equiv$' implies '$=$', $R = A$ with $x : A \in \Gamma$ for some term A.

(2) In this case $G \equiv G_1, y : C$ for some term C where $G_1 \vdash C : s$ for some $s \in \mathcal{S}$, and
$G_1 \vdash x : R$. By IH $R = A$ with $x : A \in \Gamma'$ for some term A (where $G_1 \equiv \Gamma'$ or $G \equiv G' \boxdot \Gamma'$). Hence $(x : A) \in \Gamma$ for $\Gamma \equiv \Gamma', y : C$.

(3) In this case $G \vdash x : R'$ where $R = R'$. By IH $R' = A$ with $x : A \in \Gamma$ for some term A (where $G \equiv \Gamma$ or $G \equiv G' \boxdot \Gamma$). Since $R = R'$, $R = A$.

(4) In case the last rule was a *transfer* rule, x is not a proof. Therefore this case cannot occur.

Hence in all cases: $R = A$ with $x : A \in \Gamma$ for some term A (where $G \equiv \Gamma$ or $G \equiv G' \boxdot \Gamma$).

3.4. Properties of $MPTS_\beta$

Now that we have a satisfactory Stripping Lemma, we first keep our promises from the previous chapter by proving Strong Permutation and Strengthening. Then we show that the standard properties given in [Barendregt and Hemerik 1990] for all PTSs in the Logic Cube also hold for the MPTSs in the Modal Logic Cube.

3.4.1. LEMMA. **Correctness of types**
For a generalized context G, M and A terms,

$$G \vdash M : A \Rightarrow \exists s \in \mathcal{S}[A \equiv s \vee G \vdash A : s]$$

PROOF. By induction on the derivation of $G \vdash M : A$. For the non-modal cases the proof is exactly like that for PTSs (cf. [Geuvers 1993]): the only interesting case is application where the Substitution Lemma is needed.

For $transfer_2$, $transfer_3$, and $transfer_{ax}$ the rule $transfer_1$ comes to the rescue, e. g. :

> Transfer$_2$ $G \vdash A : B$ is $G' \boxdot \varepsilon \vdash C : D$ where $G \equiv G' \boxdot \varepsilon$, an immediate consequence of $G' \vdash C : D : Type$. From this by $transfer_1$ $G' \boxdot \varepsilon \vdash D : Type$, and hence $G \vdash D : Type$ ($\exists s \in \mathcal{S}[G \vdash B : s]$).

For the import cases, $transfer_1$ is also needed:

> K-import $G \vdash A : B$ is $G' \boxdot \varepsilon \vdash \check{k}C : D$ where $G \equiv G' \boxdot \varepsilon$, an immediate consequence of $G' \vdash C : \Box D : Prop$. From this by $transfer_1$ $G' \boxdot \varepsilon \vdash \Box D : Prop$, hence by the Stripping Lemma $G' \boxdot \varepsilon \vdash D : Prop$, and so $G \vdash D : Prop$ ($\exists s \in \mathcal{S}[G \vdash B : s]$).

This is matched by the use of $refsnart_1$ in the export cases:

> K-export $G \vdash A : B$ is $G \vdash \hat{k}C : \Box D$ an immediate consequence of $G \boxdot \varepsilon \vdash C : D : Prop$. From this we obtain that $G \vdash D : Prop$ by the derived rule $refsnart_1$, hence by *boxing* $G \vdash \Box D : Prop$ ($\exists s \in \mathcal{S}[G \vdash B : s]$).

In order the prove that Strengthening is a derived rule for the MPTSs in the Modal Logic Cube, we need the following Sublemma:

3.4.2. LEMMA. **Sublemma**
For $MPTS_\beta$, if $G_1, x : A, G_2$ is a generalized context and M and B are terms, then

$$G_1, x : A, G_2 \vdash M : B \;\&\, x \notin FV(G_2, M) \;\Rightarrow\; \exists B'[B \twoheadrightarrow_\beta B' \;\&\; G_1, G_2 \vdash M : B']$$

PROOF. By induction on the derivation of $G_1, x : A, G_2 \vdash M : B$, distinguishing cases according to the last rule.

3.4.3. LEMMA. **Strengthening Lemma**
For $G_1, x : A, G_2$ a generalized context and M and B terms,

$$G_1, x : A, G_2 \vdash M : B \;\&\, x \notin FV(G_2, M, B) \;\Rightarrow\; G_1, G_2 \vdash M : B$$

PROOF. By the Sublemma we find a B' such that $B \twoheadrightarrow_\beta B'$ and $G_1, G_2 \vdash M : B$ By Correctness of Types there are two possibilities $G_1, x : A, G_2 \vdash B; s$ or $B \equiv s$ for some $s \in \mathcal{S}$. In the second case we are done, in the first case we apply the Sublemma again to $G_1, x : A, G_2 \vdash B : s$, to obtain $G_1, G_2 \vdash B : s$. Hence by *conversion* on $G_1, G_2 \vdash M : B'$, $G_1, G_2 \vdash B : s$, and $B =_\beta B'$, we have $G_1, G_2 \vdash M : B$.

3.4.4. LEMMA. **Strong Permutation Lemma**
For $G_1, x : A, y : B, G_2$ a generalized context, M and C terms with $x \notin FV(B)$,

$$G_1, x : A, y : B, G_2 \vdash M : C \Rightarrow G_1, y : B, x : A, G_2 \vdash M : C$$

PROOF. The only thing to do is to show $G_1, y : B, x : A, G_2$ is a legal context if $G_1, x : A, y : B, G_2$ is. Then we are done by Thinning ($G_1, x : A, y : B, G_2 \subseteq G_1, y : B, x : A, G_2$). By the Legality lemma we know that $G_1, x : A \vdash B : s$ for some $s \in \mathcal{S}$. By Strengthening we conclude that $G_1 \vdash B : s$, and hence that $G_1, y : B$ is a legal context. So once again by using the Legality Lemma and Thinning we derive that $G_1, y : B, x : A$ is a legal context. We can repeat this operation of applying Legality and Thinning for all declarations in G_2 and finally conclude that $G_1, y : B, x : A, G_2$ is a legal context. Note that since G_2 is a generalized context, we sometimes have to 'pass' a block (⧈). In those cases $G_1, y : B, x : A, z_1 : C_1, \ldots, z_{i-1} : C_{z-1}$ ⧈ $z_i : C_i$. By Legality we have $G_1, y : B, x : A, z_1 : C_1, \ldots, z_{i-1} : C_{z-1} \vdash C_i : s$, hence by *transfer*$_1$ $G_1, y : B, x : A, z_1 : C_1, \ldots, z_{i-1} : C_{z-1}$ ⧈ $\varepsilon \vdash C_i : s$ and so by Thinning $G_1, y : B, x : A, z_1 : C_1, \ldots, z_{i-1} : C_{z-1}$ ⧈ $z_i : C_i \vdash C_i : s$, hence the context is legal.

3.4.1. Unicity of Types

To prove Unicity of Types, we have to refer to the fact that the MPTSs in the Modal Logic Cube are 'functional'.

3.4.5. DEFINITION. A MPTS $\Box\lambda(\mathcal{S}, \mathcal{A}^{Type}, \mathcal{R})$ is *functional* if the relation $\mathcal{A}^{Type}$ is a partial function from $\mathcal{S} \times \mathcal{S}$ to $\mathcal{S}$ and:

If $s : s', s : s'' \in \mathcal{A}^{Type}$, then $s' \equiv s''$.

If $(s_1, s_2, s_3), (s_1', s_2', s_3) \in \mathcal{R}$ then $s_1 \equiv s_1'$ and $s_2 \equiv s_2'$.

3.4.6. THEOREM. *Let $\Box\lambda S$ be a functional MPTS.*
Then: if $G \vdash A : B_1$ and $G \vdash A : B_2$, $B_1 =_\beta B_2$.

PROOF. By induction on the structure of A, assuming $G \vdash A : B_i$ for $1, 2 = i$.

The functionality is needed in the case for Π:

$A \equiv \Pi x : M.N$ is a product. By the Stripping Lemma we have:

$G \vdash M : s_1$, $G, x : M \vdash N : s_2$ and $B_1 =_\beta s_3$ with $(s_1, s_2, s_3) \in \mathcal{R}$
for some $s_1, s_2, s_3 \in \mathcal{S}$, and
$G \vdash M : s_1'$, $G, x : M \vdash N : s_2'$ and $B_2 =_\beta s_3'$ with $(s_1', s_2', s_3') \in \mathcal{R}$
for some $s_1', s_2', s_3' \in \mathcal{S}$

By IH $s_1 =_\beta s_1'$ and $s_2 =_\beta s_2'$, hence $s_1 \equiv s_1'$ and $s_2 \equiv s_2'$. Then, by the assumption that $\Box\lambda S$ is singly sorted $s_3 \equiv s_3'$, and so $B_1 =_\beta s_3 =_\beta s_3' =_\beta B_2$.

Some of the modal cases require an observation on β-equality:

$A \equiv \breve{k}M$ is an import term. By the Stripping Lemma we have that for $G \equiv G' \boxdot \Gamma$, $G' \vdash M : \Box C_1$ and $C_1 =_\beta B_1$, and $G' \vdash M : \Box C_2$ and $C_2 =_\beta B_2$. By IH $\Box C_1 =_\beta \Box C_2$, but then $C_1 =_\beta C_2$, hence $B_1 =_\beta C_1 =_\beta C_2 =_\beta B_2$.

Likewise we need that if $C_1 =_\beta C_2$, then $\Box C_1 =_\beta \Box C_2$ for K-export, and that $C_1 =_\beta C_2$ implies $\neg\Box\neg C_1 =_\beta \neg\Box\neg C_2$ for B-import.

3.4.2. Subject Reduction

3.4.7. THEOREM. *If* $G \vdash A : B$ *and* $A \to_\beta A'$, *then* $G \vdash A' : B$.

PROOF. By simultaneous induction on

(i) $G \vdash A : B \,\&\, A \to_\beta A' \Rightarrow G \vdash A' : B$

(ii) $G \vdash A : B \,\&\, G \to_\beta G' \Rightarrow G' \vdash A : B$

distinguishing cases according to the last applied rule, where $G \to_\beta G'$ iff $G \equiv x_1 : A_1 \ldots \boxdot \ldots, x_n : A_n$, $G' \equiv x_1' : A_1' \ldots \boxdot \ldots, x_n' : A_n'$ and for some i one has $A_i \to_\beta A_i'$ and $A_j \equiv A_j'$ for all $j \neq i$ $(1 \leq j \leq n)$.

Proof of (i). All cases but *application* are straightforward. For the *product* and *abstraction* cases, the 'other' induction hypotheses (ii) is needed.

Application $G \vdash A : B$ is $G \vdash FN : D[x := N]$, an immediate consequence of (1) $G \vdash F : (\Pi x : C.D)$ and (2) $G \vdash N : C$. Two cases:

$A \equiv F'N' \equiv A'$ the reduction takes place inside F $(F \to_\beta F')$ or inside N $(N \to_\beta N')$. By IH on (1) and (2), $G \vdash F' : (\Pi x : C.D)$ if $F \to F'$, or $G \vdash N' : C$ if $N \to N'$. Hence $G \vdash (FN)' : D[x := N]$ since $(FN)' \equiv F'N$ or $(FN)' \equiv FN'$.

$A \equiv (\lambda x : F_1.F_2)N \equiv A'$ FN is the redex. Then $G \vdash A : B$ is $G \vdash (\lambda x : F_1.F_2)N : D[x := N]$, an immediate consequence of (1) $G \vdash \lambda x : F_1.F_2 : (\Pi x : C.D)$ and (2) $G \vdash N : C$. We have to show that $G \vdash F_2[x := N] : D[x := N]$.
By applying the Stripping Lemma to (1) we find (3) $G \vdash F_1 : s_1$ and (4) $G, x : C \vdash F_2 : D'$ for some $D' \to D$. Applying Stripping to $G \vdash (\Pi x : F_1.F_2) : s$ yields (5) $G, x : C \vdash D : s_2$ and $F_1 =_\beta C$. From this and (2) and (3), we have by *conversion* that (6) $G \vdash N : F_1$. By the Substitution Lemma and (2) we have (7) $G \vdash F_2[x := N] : D'[x := N]$ Using this lemma and (5) and (2) we can also conclude (8) $G \vdash D[x := N] : s_2$ from (5) and (2).
Since $D'[x := N] =_\beta D[x := N]$, *conversion* on (7) and (8) yields the desired conclusion $G \vdash F_2[x := N] : D[x := N]$.

Proof of ii. Here the cases where the 'other hypothesis' (i) is used are those were *start* and *weakening* are the last applied rule. The cases for *transfer*, *import*, and *export*-rules are no problem since there can be no β-redexes in empty subordinate contexts:

Transfer$_1$ $G \vdash A : B$ is $G_1 \boxdot \varepsilon \vdash C : s$ where $G \equiv G_1 \boxdot \varepsilon$, an immediate consequence of $G_1 \vdash C : s$. Since there are no redexes in ε, G_1' can only be $G_1' \boxdot \varepsilon$ (the redex is in G_1), hence by IH $G_1' \vdash C : s$. Therefore $G_1' \boxdot \varepsilon \vdash C : s$ and so $G' \vdash C : s$.

3.4.8. COROLLARY. *If* $G \vdash A : B$ *and* $B \twoheadrightarrow_\beta B'$, *then* $G \vdash A : B'$.

PROOF. Immediate, using Correctness of Types.

3.4.3. Strong normalisation

Using the erasure mapping '$|\ |$' defined earlier, which preserves β-redexes and typeability, we can prove Strong Normalisation for $MPTS_\beta$ directly from the Strong Normalisation of PTS_β ([Geuvers and Nederhof 1991]).

3.4.9. DEFINITION. Let ζ be an MPTS, $M \in Term(\zeta)$, $n \in \mathbb{N}$.

- i *n is an upperbound to the reductions starting from M* iff $\forall M_1, M_2, \ldots, M_m \in Term(\zeta).[M \to M_1 \to \ldots \to M_{m-1} \to M_m \Rightarrow m \leq n]$,
- ii *M is strongly normalizable*, or SN(M), iff $\exists n \in \mathbb{N}$. [n is an upperbound to the reductions starting from M],
- iii ζ *satisfies the strong normalization property*, or $\zeta \models$ SN, iff $\forall M \in Term(\zeta)$.SN($M$).

3.4.10. THEOREM. $\zeta \models$ SN.

PROOF. The erasure map, $|\ |$, preserves all β-redexes in M. Hence if there is an infinite reduction starting from some MPTS-term M, there is an infinite reduction starting from $|M|$ in the corresponding PTS. But this contradicts the fact that $\nu \models$ SN for all ν a PTS. Hence $\forall M \in Term(\zeta)$.SN($M$); $\zeta \models$ SN.

3.4.4. Church-Rosser for MPTS$_\beta$

The Church Rosser property of PTSs can straightforwardly be transferred to MPTSs.

3.4.11. THEOREM. *$MPTS_\beta$ has the Church Rosser property.*

PROOF. By the observation that substituting fresh variables for the occurrences of $\check{k}$, $\check{4}$, $\check{5}$, $\check{b}$ and $\hat{k}$, $\hat{d}$, $\hat{t}$ in MPTS-terms, does not change their behaviour with respect to β-reduction. This substitution operation turns MPTS terms into PTS terms and for these the Church Rosser property holds.

3.5. Properties of modal reductions

In this section we show that the annihilation rules are well-behaved. Adding the pairs of rules for $\check{k}\&\hat{k}$, $\check{4}\&\hat{t}$, and $\check{5}\&\hat{t}$ to $MPTS_\beta$, results in systems $MPTS_{\beta,annih}$ for which we can prove Subject Reduction, Strong Normalisation and Church Rosser.

3.5.1. Subject Reduction

The proof of Subject reduction for the annihilation rules is simpler than that for β-reduction, since we cannot have redexes for these rules appearing in the context.

3.5.1. THEOREM.

(*i*) *If* $G \vdash A : B$ *and* $A \rightarrow_{\hat{k}\check{k}} A'$, *then* $G \vdash A' : B$.

(*ii*) *If* $G \vdash A : B$ *and* $A \rightarrow_{\check{k}\hat{k}} A'$, *then* $G \vdash A' : B$.

(*iii*) *If* $G \vdash A : B$ *and* $A \rightarrow_{\hat{i}\check{4}} A'$, *then* $G \vdash A' : B$.

(*iv*) *If* $G \vdash A : B$ *and* $A \rightarrow_{\hat{4}\check{i}} A'$, *then* $G \vdash A' : B$.

(*v*) *If* $G \vdash A : B$ *and* $A \rightarrow_{\hat{i}\check{5}} A'$, *then* $G \vdash A' : B$.

(*vi*) *If* $G \vdash A : B$ *and* $A \rightarrow_{\hat{5}\check{i}} A'$, *then* $G \vdash A' : B$.

PROOF. By induction on the derivation.

There are only few interesting cases: for (*i*), (*iii*), and (*v*), the case of the corresponding export rule, and for (*ii*), (*iv*), and (*vi*), the case of the corresponding import rule. We show this for the $\hat{k}$&$\check{k}$-rules.

Proof of (*i*), case for *K-export*

K-export $G \vdash A : B$ is $G \vdash \hat{k}C : \Box D$ an immediate consequence of $G \boxdot \varepsilon \vdash C : D : Prop$. Two cases:

$C \rightarrow_{\hat{k}\check{k}} C'$ the $\hat{k}\check{k}$-redex is in C. By IH $G \boxdot \varepsilon \vdash C' : D : Prop$, hence $G \boxdot \varepsilon \vdash \hat{k}(C') : \Box D$. Since $\hat{k}(C') \equiv (\hat{k}C)'$, the redex is in C, $G \vdash (\hat{k}C)' : \Box D$.

$C \equiv \check{k}C_1$ the export term is the redex: $G \vdash \hat{k}(\check{k}C_1) : \Box D$ an immediate consequence of (1) $G \boxdot \varepsilon \vdash \check{k}C_1 : D : Prop$ By the Stripping on (1), (2) $G \vdash C_1 : \Box(D') : Prop$ for some term $D' =_\beta D$. But if $D' =_\beta D$, then (3) $\Box(D') =_\beta \Box D$. From (1) we have that $G \boxdot \varepsilon \vdash D : Prop$, hence by *refsnart*$_1$, $G \vdash D : Prop$ and by *boxing* (4) $G \vdash \Box D : Prop$ hence (2),(3),(4), and *conversion* yield $G \vdash C_1 : \Box D$ and so $G \vdash C' : \Box D$.

Proof of (*ii*), case for *K-import*

K-import $G \vdash A : B$ is $G_1 \boxdot \varepsilon \vdash \check{k}C : D$ where $G \equiv G_1 \boxdot \varepsilon$, an immediate consequence of $G_1 \vdash C : \Box D : Prop$. Two cases:

$C \rightarrow_{\check{k}\hat{k}} C'$ the $\check{k}\hat{k}$-redex is in C. By IH $G_1 \vdash C' : \Box D : Prop$, hence $G_1 \boxdot \varepsilon \vdash \check{k}(C') : D$. Since $\check{k}(C') \equiv (\check{k}C)'$, the redex is in C, $G \vdash (\check{k}C)' : \Box D$.

$C \equiv \hat{k}C_1$ the import term is the redex: $G_1 \boxdot \varepsilon \vdash \check{k}(\hat{k}C_1) : D$ an immediate consequence of (1) $G_1 \vdash \hat{k}C_1 : \Box D : Prop$. By Stripping on (1), (2) $G_1 \boxdot \varepsilon \vdash C_1 : D' : Prop$ for some term (3) $D' =_\beta D$. From (1), we have that (4) $G_1 \vdash \Box D : Prop$, and hence by *Transfer*$_1$ that (5) $G_1 \boxdot \varepsilon \vdash \Box D : Prop$. Applying Stripping then yields (6) $G_1 \boxdot \varepsilon \vdash D : Prop$. By *conversion* on (2),(3),(6), $G_1 \boxdot \varepsilon \vdash C_1 : D$ and so $G \vdash C' : D$.

3.5.2. Strong Normalisation

The idea in proving Strong Normalisation for the modal reductions is to map the annihilation-reductions of the MPTS-terms to β-reductions in the corresponding PTS-terms. In this way any infinite reduction path starting from a MPTS-term, be it a β-, annihilation-, or mixed chain, will be mapped to an infinite chain of β-reductions from the corresponding PTS term. The proof uses a variation on the erasure mapping defined earlier, where the 'hat' and 'check' functions are mapped to identity.

3.5.2. DEFINITION. **Erasure mapping**
Let $|\ |$ be a mapping of MPTS-terms to PTS-terms, which erases the modal terms:

i $|\Box A| = |A|\ , |\neg\Box\neg A| = |A|$

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 $|\Gamma, x : A| = |\Gamma|, |x : A|$, $|G \boxdot \Gamma| = |G|, |\Gamma|$

iv $|A : B| = |A| : |B|$, $|\varepsilon| = \varepsilon$, $|x| = x$ (for $x \in Var$), $|s| = s$ (for $s \in \mathcal{S}$)

v $|\check{k}A| = (\lambda x : |B|.x)|A|$, $|\check{b}A| = (\lambda x : |B|.x)|A|$, $|\check{4}A| = (\lambda x : |B|.x)|A|$,
$|\check{5}A| = (\lambda x : |B|.x)|A|$, where B is a type of A, and $x \notin FV(A \cup B)$.
$|\hat{k}A| = (\lambda x : |B|.x)|A|$, $|\hat{d}A| = (\lambda x : |B|.x)|A|$, $|\hat{t}A| = (\lambda x : |B|.x)|A|$
where B is a type of A, and $x \notin FV(A \cup B)$.

For this mapping we can prove that it preserves β-reduction and typeability in the same way as before.

3.5.3. LEMMA. **Preservation of substitution**
$|\ |$ preserves substitutions: for MPTS-terms A and B $|A|[x := |B|] = |A[x := B]|$.

PROOF. By induction on the structure of the term A.

3.5.4. LEMMA. **Preservation of β-reduction**
$|\ |$ preserves β-reductions: for MPTS-terms A, M and N, if $(\lambda x : A.M)N \to_\beta M[x := N]$ then
$|(\lambda x : A.M)N| \to_\beta |M[x := N]|$.

PROOF. $|(\lambda x : A.M)N| = |(\lambda x : A.M)||N| = (\lambda x : |A|.|M|)|N| \to_\beta |M|[x := |N|]$, by the Substitution Preservation Lemma $|M|[x := |N|] = |M[x := N]|$.

3.5.5. LEMMA. **Preservation of typeability**
$|\ |$ preserves typeability:
$G \vdash A : B \Rightarrow |G| \vdash |A| : |B|$.

PROOF. By induction on the derivation of $G \vdash A : B$.

The crucial part of the proof is to show that if a term M reduces to M' in one annihilation step, $|M|$ reduces to $|M'|$ in *one or more* steps of β-reduction ($\twoheadrightarrow_\beta$).

3.5.6. LEMMA. **Preservation of annihilations** $|\ |$ *preserves annihilation reductions.*

(i) *If* $\hat{k}\check{k}A \to_{\hat{t}\check{4}} A$, *then* $|\hat{k}\check{k}A| \twoheadrightarrow_\beta |A|$,
If $\hat{t}\check{4}A \to_{\hat{t}\check{4}} A$, *then* $|\hat{t}\check{4}A| \twoheadrightarrow_\beta |A|$,
If $\hat{t}\check{5}A \to_{\hat{t}\check{5}} A$, *then* $|\hat{t}\check{5}A| \twoheadrightarrow_\beta |A|$.

(ii) *If* $\check{k}\hat{k}A \to_{\check{4}\hat{t}} A$, *then* $|\check{k}\hat{k}A| \twoheadrightarrow_\beta |A|$
If $\check{4}\hat{t}A \to_{\check{4}\hat{t}} A$, *then* $|\check{4}\hat{t}A| \twoheadrightarrow_\beta |A|$
If $\check{5}\hat{t}A \to_{\check{5}\hat{t}} A$, *then* $|\check{5}\hat{t}A| \twoheadrightarrow_\beta |A|$

PROOF. (i) $M \equiv \hat{k}\check{k}A$ or $M \equiv \hat{t}\check{4}A$ or $M \equiv \hat{t}\check{5}A$
$|\hat{k}\check{k}A| = (\lambda x : |B|.x)|\check{k}A| =_{(\alpha)} (\lambda x : |B|.x)((\lambda y : |B|.y)|A|)$,
$|\hat{t}\check{4}A| = (\lambda x : |B|.x)|\check{4}A| =_{(\alpha)} (\lambda x : |B|.x)((\lambda y : |B|.y)|A|)$,
$|\hat{t}\check{5}A| = (\lambda x : |B|.x)|\check{5}A| =_{(\alpha)} (\lambda x : |B|.x)((\lambda y : |B|.y)|A|)$.
In all cases $M = (\lambda x : |B|.x)((\lambda y : |B|.y)|A|)$, this term can be reduced in two ways:

$$(\lambda x : |B|.x)((\lambda y : |B|.y)|A|) \to_\beta (\lambda y : |B|.y)|A| \to_\beta |A|$$

$$(\lambda x : |B|.x)((\lambda y : |B|.y)|A|) \to_\beta (\lambda x : |B|.x)|A| \to_\beta |A|$$

In all cases $|\hat{k}\check{k}A|$, $|\hat{t}\check{4}A|$ and $|\hat{t}\check{5}A|$ reduce to $|A| \equiv |M'|$.

(ii) $M \equiv \check{k}\hat{k}A]$ or $M \equiv \check{4}\hat{t}A$ or $M \equiv \check{5}\hat{t}A$
$|\check{k}\hat{k}A| = (\lambda x : |B|.x)|\hat{k}A| =_{(\alpha)} (\lambda x : |B|.x)((\lambda y : |B|.y)|A|)$,
$|\check{4}\hat{t}A| = (\lambda x : |B|.x)|\hat{t}A| =_{(\alpha)} (\lambda x : |B|.x)((\lambda y : |B|.y)|A|)$,
$|\check{5}\hat{t}A| = (\lambda x : |B|.x)|\hat{t}A| =_{(\alpha)} (\lambda x : |B|.x)((\lambda y : |B|.y)|A|)$.
In all cases $M = (\lambda x : |B|.x)((\lambda y : |B|.y)|A|)$, this term can be reduced in two ways:

$$(\lambda x : |B|.x)((\lambda y : |B|.y)|A|) \to_\beta (\lambda y : |B|.y)|A| \to_\beta |A|$$

$$(\lambda x : |B|.x)((\lambda y : |B|.y)|A|) \to_\beta (\lambda x : |B|.x)|A| \to_\beta |A|$$

In all cases $|\check{k}\hat{k}A|$, $|\check{4}\hat{t}A|$ and $|\check{5}\hat{t}A|$ reduce to $|A| \equiv |M'|$.

Now we can prove Strong Normalisation using the same argument as for $MPTS_\beta$.

3.5.7. THEOREM. *Let* ζ *be an* $MPTS_{\beta,annih}$, *then:*

$$\zeta \models \mathrm{SN}$$

PROOF. The erasure map, $|\ |$, preserves all β- and annihilation redexes in M. Hence if there is an infinite reduction starting from some MPTS-term M, there is an infinite (β)-reduction starting from $|M|$ in the corresponding PTS. But this contradicts the fact that $\nu \models$ SN for all ν a PTS ([Geuvers and Nederhof 1991]). Hence $\forall M \in Term(\zeta).\mathrm{SN}(M)$; $\zeta \models$ SN.

3.5.3. Church-Rosser

We prove Church Rosser for $MPTS_{\beta,annih}$ by showing the Church Rosser property for each of the combinations of β-reduction with an annihilation pair separately. From this we can prove Church Rosser for the entire systems with all annihilation rules by arguing that there are no 'critical pairs'.

3.5.8. THEOREM.

$MPTS_{\beta,\check{k}\&\hat{k}}$ has the Church Rosser property

$MPTS_{\beta,\check{4}\&\hat{i}}$ has the Church Rosser property

$MPTS_{\beta,\bar{5}\&\bar{i}}$ has the Church Rosser property

We show the proof for (i), the other cases are similar. To prove CR for MPTS with β and $\check{k}\hat{k}$, $\hat{k}\check{k}$-reductions, we use a Martin-Löf-Tait-type proof as in [Hindley and Seldin 1986]. We first give a sketch of the proof:

> Give a definition of residuals and minimal complete development (mcd) that includes $\check{k}\hat{k}$ and $\hat{k}\check{k}$-reduction steps.
>
> Lemma 1: If $M >_{mcd} M'$ and $N >_{mcd} N'$ then $M[x := N] >_{mcd} M'[x := N']$
>
> Lemma 2: $P >_{mcd} A$ and $P >_{mcd} B$ then $\exists T : A >_{mcd} T, B >_{mcd} T$
>
> From Lemma 2 by induction on the number of $\beta - \check{k}\hat{k} - \hat{k}\check{k}$-steps $P >_{mcd} M$ and $P >_{\beta,\check{k}\&\hat{k}} N$ then $\exists T : M >_{\beta,\check{k}\&\hat{k}} T, N >_{mcd} T$.
>
> Given the observation that a single $\beta-$, $\check{k}\hat{k}-$, or $\hat{k}\check{k}-$ step is an $>_{mcd}$-step we have $P >_{1(\beta,\check{k}\&\hat{k})} M$ and $P >_{\beta,\check{k}\&\hat{k}} N$ then $\exists T : M >_{\beta,\check{k}\&\hat{k}} T, N >_{\beta,\check{k}\&\hat{k}} T$.
>
> Hence by induction on the number of $\beta-$, $\check{k} - \&\hat{k}$-steps from P to N: $P >_{\beta,\check{k}\&\hat{k}} M$ and $P >_{\beta,\check{k}\&\hat{k}} N$ then $\exists T : M >_{\beta,\check{k}\&\hat{k}} T, N >_{\beta,\check{k}\&\hat{k}} T$.

3.5.9. DEFINITION. **Substitution**
For any M, N, x, define $M[x := N]$ to be the result of substituting N for every free occurrence of x in M, and changing bound variables to avoid clashes. The definition is by induction on M.

(a) $x[x := N] \equiv N$

(b) $a[x := N] \equiv a$

(c) $PQ[x := N] \equiv (P[x := N]Q[x := N])$

(d) $\lambda x.P[x := N] \equiv \lambda x.P$

(e) $(\lambda y.P)[x := N] \equiv \lambda y.(P[x := N])$ if $y \not\equiv x$ and $y \notin FV(N)$ or $x \notin FV(P)$

(f) $(\lambda y.P)[x := N] \equiv \lambda z.(P[x := N][z := y])$ if $y \not\equiv x$ and $y \in FV(N)$ or $x \in FV(P)$

(g) $(\check{k}P)[x := N] \equiv \check{k}(P[x := N]), (\check{4}P)[x := N] \equiv \check{4}(P[x := N]),$
$(\check{5}P)[x := N] \equiv \check{5}(P[x := N]), (\check{b}P)[x := N] \equiv \check{b}(P[x := N]),$
$(\hat{k}P)[x := N] \equiv \hat{k}(P[x := N]), (\hat{d}P)[x := N] \equiv \hat{d}(P[x := N]),$
$(\hat{t}P)[x := N] \equiv \hat{t}(P[x := N]).$

In (f), z is chosen to be the first variable $\notin FV(NP)$

3.5.10. DEFINITION. **Contractions and Reductions**

Contraction: an ordered triple $< X, Y, R >$ where X is a term, R is an occurrence of a redex in X and Y is the result of contracting R in X.

$$X >_R Y$$

Reduction: ρ is a finite or infinite series of contractions of the form $X_1 >_{R_1} X_2 >_{R_2} X_3 >_{R_3} \cdots$.

Length: of ρ is the number of its contractions, (finite or infinite). If *lenght*(ρ) is finite, say n, then X_{n+1} is called the *terminus* of ρ.

3.5.11. DEFINITION. **Reduction rules**

α-reduction: is the change of a single bound variable. Let a term P contain an occurrence of $\lambda x.M$, and let $y \notin FV(M)$. The act of replacing this $\lambda x.M$ by $\lambda y.Mx := y]$ is called 'a change of bound variable' in P.

β-reduction: any term of the form $(\lambda x.M)N$ is called a β -redex and the corresponding term $M[x := N]$ is called its contractum. If a term P contains an occurrence of $(\lambda x.M)N$ and we replace that occurrence by $M[x := N]$ and the result is P', we say that we have 'contracted' the redex-occurrence in P, or $P >_\beta P'$. We say P β-reduces to Q, or $P >_\beta Q$ iff Q is obtained from P by a finite (perhaps empty) series of β-contractions and changes of bound variables.

$\hat{k}\check{k}$-reduction: any term of the form $\hat{k}(\check{k}M)$ is called a $\hat{k}\check{k}$-redex and M is its contraction.

$\check{k}\hat{k}$-reduction: Any term of the form $\check{k}(\hat{k}M)$ is called a $\check{k}\hat{k}$-redex and M is its contraction.

The first step in proving CR for combined β- and $\check{k}\&\hat{k}$- reductions is to extend the definition of residuals and mcd's. The definition of residuals describes what happens to a redex S in a term P when another redex R in P is contracted. (Of its cases only 1,2, are actually needed in the proof (these are the simple cases); but 4 is often used elsewhere in reduction theory, so it is included here).

3.5.12. DEFINITION. **Residuals**
Let R, S be $\beta-$, $\hat{k}\check{k}-$, $\check{k}\hat{k}$-redexes in an MPTS-term P. When R is contracted, let P change to P'. The *residuals* of S with respect to R are redexes in P' defined as follows

Case 1: R, S are *non-overlapping* parts of P. Then contracting R leaves S unchanged. This unchanged S in P' is called the residual of S.

Case 2: $R \equiv S$. Then contracting R is the same as contracting S. We say S has no residual in P'.

Case 3: R is *part of* S *and* $R \not\equiv S$. Then S has form

$(\lambda x.M)N$ and R is in M or in N. Contracting R changes M to M' and N to N' and S to $(\lambda x.M')N$ or $(\lambda x.M)N'$ in P', this is the residual of S.

$\hat{k}(\check{k}M)$ and R is in M. Contracting R changes M to M' and S to $\hat{k}(\check{k}M')$ in P', this is the residual of S.

$\check{k}(\hat{k}M)$ and R is in M. Contracting R changes M to M' and S to $\check{k}(\hat{k}M')$ in P', this is the residual of S.

Case 4: S *is part of* R *and* $S \not\equiv R$. Then S has form

$(\lambda x.M)N$ and R is in M or in N. Contracting R changes $(\lambda x.M)N$ to $M[x := N]$.

S is in M When $M[x := N]$ is formed from M, then S is changed to a redex S' with one of the forms $S[x := N], \ldots, S[x := N][y_1 := z_1] \ldots [y_n := z_n], S$, depending on how many times clause (f) is used in determining $M[x := N]$ and whether S is in the scope of a λx in M. This S' is called the residual of S (it is clearly a β-redex)

S is in N When $M[x := N]$ is formed, there is an occurrence of S in each substituted N. These are called the residuals of S.

$\hat{k}(\check{k}M)$ and R is in M. Contracting R changes $\hat{k}(\check{k}M)$ to M. When M is formed from $\hat{k}(\check{k}M)$, S does not change. Hence S is called the residual of S.

$\check{k}(\hat{k}M)$ and R is in M. Contracting R changes $\check{k}(\hat{k}M$ to M. When M is formed from $\check{k}(\hat{k}M$, S does not change. Hence S is called the residual of S.

Note that except in the subcase of 4 where S is in M, S has at most one residual.

3.5.13. DEFINITION. **MCD's**

Let $R_1, \ldots, R_n$ $(n \geq 0)$ be redexes in a term P. An R_i is called *minimal* iff it properly contains no other R_j. We say

$$P >_{mcd} Q$$

iff Q is obtained from$P$ by the following process called a *minimal complete development* (mcd) of the set $\{R_1, \ldots, R_n\}$. First contract any minimal R_i (say $i = 1$ for convenience). By the note above, this leaves at most $n-1$ residuals $R'_2, \ldots, R'_n$ of $R_2, \ldots, R_n$. Contract any minimal R'_j. This leaves at most $n-2$ residuals. repeat this until no residuals are left. Then make as many α-contractions as you like. (This process is not unique.)

3.5.14. FACTS.

(a) In any non-empty set of redexes, there is always a minimal member.

(b) If $n = 0$, an mcd is perhaps just a empty series of α-steps.

(c) A single β-, $\hat{k}\check{k}$-, or $\check{k}\hat{k}$- contraction is a mcd of a one-member set.

(d) Non-mcd's exist; for example the reduction

$$(\lambda x.xy)(\lambda z.z) >_{1\beta} (\lambda z.z)y >_{1\beta} y.$$

(e) The relation $>_{mcd}$ is not transitive; e.g. in (d) there is clearly no mcd from $(\lambda x.xy)(\lambda z.z)$ to y.

(f) If $M >_{mcd} M'$ and $N >_{mcd} N'$, then $MN >_{mcd} M'N'$.

(g) It is fairly easy to show that, modulo congruence. Q is determined uniquely by the set $\{R_1, \dots R_n\}$. (this fact will not be needed here, however.)

3.5.15. LEMMA. *If $P >_{mcd} Q$ and $P \equiv_\alpha P^*$, then $P^* >_{mcd} Q$.*

PROOF. For just β-reduction this is proven in [Hindley and Seldin 1986] by 'a boring induction on the number of β-steps from P to Q'. To show that the possible occurence of $\hat{k}\&\check{k}$-reductions will not really complicate the proof, the following observation suffices: If $P \equiv_\alpha P^*$ and $P >_{\hat{k}\check{k}} Q$ ($P >_{\check{k}\hat{k}} Q$) ,then $P \equiv \hat{k}(\check{k}M)$ for some term M. Hence if $P \equiv_\alpha P^*$, $P^* \equiv \hat{k}\check{k}M^*(\check{k}(\hat{k}M^*))$, for some $M^* \equiv_\alpha M$. Since $M^* \equiv Q$ in $P^* >_{\hat{k}\check{k}} Q$ and $M \equiv Q$ in $P >_{\hat{k}\check{k}} Q$, we can obtain the original Q (M) by an mcd from the α-reduced P^*, by simply executing the reverse α-contraction from M^* to M.

3.5.16. LEMMA. *If $M >_{mcd} M'$ and $N >_{mcd} N'$ then $M[x := N] >_{mcd} M'[x := N']$.*

PROOF. by the previous lemma and the the fact that α-equivalence is preserved under substitution, we may assume that no variable bound in M is free in xM, and that the given mcd's have no α-steps. We proceed by induction on M. Let $R_1, \dots, R_n$ be the redexes developed in the given mcd of M.

3.5.17. LEMMA. *If $P >_{mcd} A$ and $P >_{mcd} B$ then $\exists T(A >_{mcd} T, B >_{mcd} T$.*

PROOF. By induction on (the structure of) P, where we may assume that the given mcd's have no α-steps because of lemma 3.5.14.

3.5.18. THEOREM. *CR holds for MPTS-terms and $>_{\beta,\hat{k}\&\check{k}}$.*

PROOF. Let $P >_{\beta,\hat{k}\&\check{k}} M$ and $P >_{\beta,\hat{k}\&\check{k}} N$. We must find a T such that $M >_{\beta,\hat{k}\&\check{k}} T$ and $N >_{\beta,\hat{k}\&\check{k}} T$.

By induction on the length of the reduction from P to M, it is enough to prove

(1) If $P >_{1\beta,\hat{k}\&\check{k}} M$ and $P >_{\beta,\hat{k}\&\check{k}} N$, then $\exists T : (M >_{\beta,\hat{k}\&\check{k}} T, N >_{\beta,\hat{k}\&\check{k}} T)$.

Given that a single $\beta-$, $\check{k}\hat{k}-$, or $\hat{k}\check{k}-$ step is a $>_{mcd}$-step, 91) follows from

(2) If $P >_{mcd} M$ and $P >_{\beta,\hat{k}\&\check{k}} N$, then $\exists T : M >_{\beta,\hat{k}\&\check{k}} T, N >_{mcd} T$.

But (2) comes from lemma 3.5.16. by induction on the number of $\beta - \check{k}\hat{k} - \hat{k}\check{k}$-steps.

3.5.19. COROLLARY. *$MPTS_{\beta,annih}$ has the Church Rosser property.*

PROOF. By the theorem above and the observation that an $MPTS_{\beta,\breve{k}\&\hat{k},\ \breve{4}\&\hat{\imath},\ \breve{5}\&\hat{\imath}}$ can have only 'trivial' critical pairs: there can be terms which are reducable by different annihilation rules, but all reductions lead to the same term in one step. These (sub)terms are $\breve{k}(\hat{k}(\breve{k}M))$, $\breve{4}(\hat{\imath}(\breve{4}M))$, $\breve{5}(\hat{\imath}(\breve{5}M))$, and $\hat{k}(\breve{k}(\hat{k}M))$, $\hat{\imath}(\breve{4}(\hat{\imath}M))$, $\hat{\imath}(\breve{5}(\hat{\imath}M))$. We show the 'triviality' of $\breve{k}(\hat{k}(\breve{k}M))$ and $\hat{k}(\breve{k}(\hat{k}M))$, the other cases are completely analogous.

> $\breve{k}(\hat{k}(\breve{k}M))$ can be reduced by applying $\breve{k}\hat{k}$-reduction , as well as $\hat{k}\breve{k}$-reduction (the redexes are underlined):
>
> $$\underline{\breve{k}(\hat{k}}(\breve{k}M)) \rightarrow_{\breve{k}\hat{k}} \breve{k}M$$
> $$\breve{k}\underline{(\hat{k}(\breve{k}}M)) \rightarrow_{\hat{k}\breve{k}} \breve{k}M$$
>
> Hence both reductions lead to the same term.
>
> $\hat{k}(\breve{k}(\hat{k}M))$ can be reduced by applying $\hat{k}\breve{k}$-reduction , as well as $\breve{k}\hat{k}$-reduction (the redexes are underlined):
>
> $$\underline{\hat{k}(\breve{k}}(\hat{k}M)) \rightarrow_{\hat{k}\breve{k}} \hat{k}M$$
> $$\hat{k}\underline{(\breve{k}(\hat{k}}M)) \rightarrow_{\breve{k}\hat{k}} \hat{k}M$$
>
> Hence both reductions lead to the same term.

Note that the only potential critical pairs that are not trivial, $\hat{4}(\breve{\imath}(\breve{5}M))$ and $\breve{5}(\breve{\imath}(\hat{4}M))$, are not typeable in any $MPTS_{\beta}$.

Chapter 4

More agents, more modalities

In the previous chapters we have considered intensional logics with one modal operator used by one reasoning agent. However, from the perspective of (information) dialogues intensional logic will have to deal with at least two agents and have more than one modal operator, reflecting different levels of faith an agent may have in his information.

This chapter examines the technical possibilities for extensions of the systems treated before in these directions, based on the work of Wiebe van der Hoek ([Van der Hoek 1992]) on the system KB_{CD} orginally proposed by Kraus and Lehmann ([Kraus and Lehmann 1986]). We will investigate if and how his, modeltheoretic, approach transfers to Fitch-style natural deduction and from there to MPTS's.

4.1. Natural deduction

In the system KB_{CD} the modal operator $\Box$ is interpreted 'epistemically', it has operators for 'knowledge' and 'belief': $K_a\varphi$ meaning 'Agent a knows that φ' and $B_a\varphi$ meaning 'Agent a believes φ'. Besides these individual notions the system also deals with knowledge and belief concerning the group as a whole, $C\varphi$ meaning 'It is Common Knowledge that φ' and $D\varphi$ representing 'It is Common Belief that φ'. The properties of these operators and their interactions are captured in the following axiomatization (where operators E and F stand for respectively 'everybody knows' and 'everybody believes'):

A0) Any axiomatization of the propositional calculus.

R0) $\vdash \varphi, \varphi \supset \psi \Rightarrow \psi$

A1) $K_a(\varphi \supset \psi) \supset (K_a\varphi \supset K_a\psi)$

A2) $K_a\varphi \supset \varphi$

A3) $\neg K_a\varphi \supset K_a\neg K_a\varphi$

A4) $C(\varphi \supset \psi) \supset (C\varphi \supset C\psi)$

A5) $C\varphi \supset E\varphi$

A6) $C\varphi \supset EC\varphi$

A7) $C(\varphi \supset E\varphi) \supset (\varphi \supset C\varphi)$

R1) $\vdash \varphi \Rightarrow \vdash C\varphi$

A8) $B_a(\varphi \supset \psi) \supset (B_a\varphi \supset B_a\psi)$

A9) $\neg B_a \bot$

A10) $D(\varphi \supset \psi) \supset (D\varphi \supset D\psi)$

A11) $D\varphi \supset F\varphi$

A12) $D\varphi \supset FD\varphi$

A13) $D(\varphi \supset F\varphi) \supset (F\varphi \supset D\varphi)$

A14) $K_a\varphi \supset B_a\varphi$

A15) $B_a\varphi \supset K_a B_a\varphi$

A16) $C\varphi \supset D\varphi$

In the following sections different parts of this system will be used to study different aspects of multi-modal multi-agent systems. First we look into generalizing the Fitch-style deduction to multi-agent and multi-modal systems concentrating on reasoning about knowledge (K_a) and belief (B_a) for a group of agents. Then we investigate interaction between modalities (and agents) using the axioms relating knowledge and belief. Finally group modalities are considered by means of Common knowledge and Common Belief.

4.1.1. Multi-agent Multi-modal logic

To formulate modal logic for groups of agents, we introduce an index set 'People' $= \{1, \dots, m\}$, following [Van der Hoek 1992]. Each of these agents has his own 'copy' of the modal operator: $\Box_1, \dots, \Box_m$. A modal logic for m agents is considered normal if it is closed under the following indexed normality rule:

$$\textit{Normality} \quad \frac{(\varphi_1 \wedge \dots \wedge \varphi_n) \supset \psi}{(\Box_a\varphi_1 \wedge \dots \wedge \Box_a\varphi_n) \supset \Box_a\psi} \text{ for } a = 1, \dots m \ (n \geq 1)$$

The smallest normal logic $K_{(m)}$, has just this rule. It can alternatively be characterized as the set of propositions derivable by means of:

- all propositional tautologies
- axiom: $\Box_a(\varphi \supset \psi) \supset (\Box_a\varphi \supset \Box_a\psi)$ for $a = 1, \dots m$
- rules:

$$\textit{Modus Ponens} \quad \frac{\varphi \quad \varphi \supset \psi}{\psi}$$

$$\textit{Generalization}: \textit{ if } \varphi \textit{ is a thesis, then } \Box_a\varphi \textit{ is a thesis } (a = 1, \dots, m)$$

Like before we consider normal systems resulting from the extension of $K_{(m)}$ now with 'indexed' versions of the familiar axiom(schema)s:

$D \ : \Box_a\varphi \supset \neg\Box_a\neg\varphi \ (a = 1, \ldots m)$

$T \ : \Box_a\varphi \supset \varphi \ (a = 1, \ldots m)$

$B \ : \varphi \supset \Box_a\neg\Box_a\neg\varphi \ (a = 1, \ldots m)$

$4 \ : \Box_a\varphi \supset \Box_a\Box_a\varphi \ (a = 1, \ldots m)$

$5 \ : \neg\Box_a\varphi \supset \Box_a\neg\Box_a\varphi \ (a = 1, \ldots m)$

The following convention will be used in referring to the normal extensions of $K_{(m)}$: $(KS_1 \ldots S_n)_{(m)}$ = the smallest normal system of modal logic for m agents containing (every instance of) the schemas $S_1 \ldots S_n$.

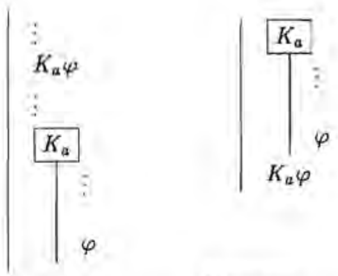
Of course the above formulation of the modal logics already suggests how we can extend the natural deduction system to more agents, but we will develop this idea in the setting of the system KB_{CD}.

In epistemic logic the modal operator $\Box$ is read as a 'knowledge' operator K, meaning 'it is known that ...', or in a logic with more agents K_a 'agent a knows that ...'. Using this operator principles of reasoning about knowledge can be expressed formally. KB_{CD} has the following:

> $K_a(\varphi \supset \psi) \supset (K_a\varphi \supset K_a\psi)$ (A1), which means that a reasoner knows the logical consequences of his knowledge
>
> $K_a\varphi \supset \varphi$ (A2), 'veridicality': if an agent knows φ, then φ is the case.
>
> $K_a\varphi \supset K_aK_a\varphi$ (consequence of A1, A2 and A3), 'positive introspection': if an agent knows φ he *knows* that he knows φ.
>
> $\neg K_a\varphi \supset K_a\neg K_a\varphi$ (A3) 'negative introspection': if an agent does not know φ, he *knows* that he does not know φ.

Hence each of the agents has a $KT45$ logic for his particular knowledge operator, for which we could give a natural deduction system.

We can unite the structurally identical deduction systems for all individual knowledge operators in a more general formulation of the usual deduction rules. Each agent a has his own operator, that is introduced or eliminated by means of the subordinate K_a-proofs. The import and export rules now only apply if the index of the subordinate K-proof and the agent-index operator are identical. For instance for the basic import rule we have to demand that only formulas of the general form $K_a\varphi$ where a is the index of the K-subordinate proof may be imported:



K-import K-export

Likewise a conclusion (φ) from a strict categorical subordinate K_a-proof may only be brought back to the main proof under the knowledge operator with index a.

In terms of Kripke semantics the condition that index of the operator and the strict subordinate proof match prohibits that the epistemic alternatives of different agents are 'confused'. Each of the knowledge operators K_a corresponds to an accessibility relation R_a which has as its extension all the epistemic alternatives of agent a (all the descriptions of the world that are still possible with respect to agent a's knowledge). Given the analogy between strict subordinate proofs and possible worlds, a strict subordinate K_a-proof corresponds to an arbitrary epistemic alternative of agent a. Therefore one has to make sure that only propositions that are known by a (and hence hold in each of his epistemic alternatives) are imported. A conclusion derived in a categorical strict subordinate K_a-proof is derivable in all epistemic alternatives of a, and hence should be exported as knowledge of a.

By 'indexing' the additional import and export rules in the same way, we obtain a natural deduction system for $KT45_{(m)}$. This simple system already fulfills the basic requirements for reasoning about knowledge of other agents as can be seen from the following example.

4.1.1. EXAMPLE.

1.	$K_1K_2K_3\varphi$	
	$\boxed{K_1}$	
2.	$K_2K_3\varphi$	(K-import 1)
	$\boxed{K_2}$	
3.	$K_3\varphi$	(K-import 2)
	$\boxed{K_3}$	
4.	φ	(K-import 3)
5.	φ $(K_3\varphi)$	(T/K-export 4)
6.	φ $(K_3\varphi$ $K_2\varphi$ $K_2K_3\varphi)$	(T/K-export 5)
7.	φ $(K_3\varphi$ $K_2\varphi$ $K_2K_3\varphi$ $K_1\varphi$ $K_1K_2\varphi$ $K_1K_3\varphi$ $K_1K_2K_3\varphi)$	(T/K-export 6)

Suppose that one has the information that $K_1K_2K_3\varphi$: 'Agent 1 knows that agent 2 knows that agent 3 knows that φ'. From this it should follow that φ is the case by the 'veridicality' of knowledge. We can indeed reach this conclusion by using the newly defined import and export rules. Given the formula $K_1K_2K_3\varphi$ we are allowed to start a K_1 subordinate proof where $K_2K_3\varphi$ holds by K-import. Intuitively this means switching the perspective of the deduction to an arbitrary epistemic alternative of agent 1. There 'pretending to be agent 1' we find that $K_2K_3\varphi$ holds (agent 1 knows this). In this way we can proceed by opening a strict subordinate K_2-proof, and then a subordinate K_3-proof where we find that φ. This conclusion can then be brought back through the subordinate proofs by the T-export rule (there are no assumptions involved).

The example shows that using only combinations of K- and T-rules all 'projections' out of a row of knowledge operators can be obtained: from $K_1 \ldots K_n\varphi$ one can conclude the fact (φ), knowledge of this fact of all agents involved ($K_a\varphi$ $a \in \{1,\ldots,n\}$), and all nestings (substrings) which respect the order of the original row: first 'strip' the formula φ of all modal operators, and then add the wanted operators (K_a) back on by using K-export out of the corresponding (a) subordinate proof, and T-export (leaving the formula unchanged) in the other cases. Note that these modal rules do not allow us to 'switch operators': we cannot derive $K_1K_3K_2\varphi$.

Multi-modal logic

In epistemic logic the motivation for a multi-modal logic is usually that one wants to be able to express different 'degrees of certainty' with respect to one's propositions, or in terms of multi-agent logics allow agents to have different levels of 'faith' in their information.

Knowledge represents one extreme on the scale of certainty: if an agents knows that something is the case in reality, this is expressed in the 'veridicality axiom' $K_a\varphi \supset \varphi$. Therefore this notion is often contrasted with that of 'belief' which is fundamentally weaker in the sense that believing something does not imply that it is true. For belief we demand no more than 'consistency': you can not believe a proposition and its negation at the same time $B_a\varphi \supset \neg B_a\neg\varphi$.

Given the consistency demand, a lot of intuitively plausible axiomatizations for the belief operator remain possible, however we will not go into the interesting discussion as to which is the right one. Instead we will follow the system KB_{CD} in adopting the logic $(KD45)_{(m)}$ for belief, hence for each agent a ($\in$ *People*) the following axiom(scheme)s hold:

- K $B_a(\varphi \supset \psi) \supset (B_a\varphi \supset B_a\psi)$ 'logical consequence'
- D $B_a\varphi \supset \neg B_a\neg\varphi$ 'consistency of belief'
- 4 $B_a\varphi \supset B_aB_a\varphi$ 'positive introspection'
- 5 $\neg B_a\varphi \supset B_a\neg B_a\varphi$ 'negative introspection'

Deductively this means that we now add strict subordinate B_a-proofs and import and export rules for formulas containing B_a-operators. Instead of having separate definitions of the rules for knowledge and the rules for belief, we unify the multi-agent logics notationally by introducing an additional 'modality-index' on the operators and the subordinate proofs.

Fitch-style deduction for multi-agent multi-modal systems

A formal definition of the deduction system $\Box PROP_{fitch}^{People,Modalities}$ for more agents and modalities is obtained from the definition of $\Box PROP_{fitch}$ by giving each of the agents ($\in$ *People*) and operators ($\in$ *Modalities*) their own *strict* subordinate proofs and corresponding import and export rules.

4.1.2. DEFINITION. **Proof figure**
A *proof figure* **D** is a mathematical structure consisting of:

1 a closed interval $D = [1, n]$, where $D \subset IN$,

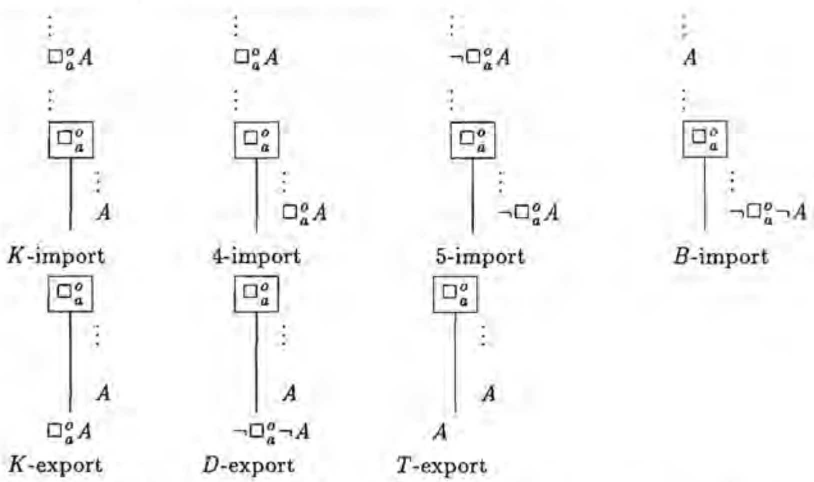
2 a function $F : D \rightarrow PROP$, and

3 a collection **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 **I** of subintervals is the union of two disjoint subcollections **H** and **M**:

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

M the *modal intervals* of the proof figure, this set is the union of all sets $\mathbf{M}^o_a$ where $o \in Modalities$, the set of operator indices ($Modalities = \{1, \ldots, n\}$ for some $n \in IN$) and $a \in People$, the set of operator indices ($People = \{1, \ldots, n\}$ for some $n \in IN$). D may not be an element of **M**.

4.1.3. DEFINITION. **Deduction rules**
The rules for the propositional connectives are as before, the modal rules have to be formulated with respect to the agent and operator index.



4.1.4. DEFINITION. **Application of deduction rules**
Given a proof figure **D**, with interval $D = [1, n]$, formulas $F_1, \ldots, F_n$ and intervals **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:

6 $R = K$ *import*.
If the premiss $\Box^o_a A$ lies in interval $I \in \mathbf{I}$ where $o \in Modalities$, $a \in People$ and the conclusion $E = A$ lies in the interval $J \in \mathbf{M}^o_a$, then it has to be the case that *the interval* J lies in *the interval* I.

7 $R = 4\,import$.
If the premiss $\Box^o_a A$ lies in interval $I \in \mathbf{I}$ where $o \in O_{4\,import} (\subseteq Modalities)$, $a \in People$ and the conclusion $E = \Box^o_a A$ lies in the interval $J \in \mathbf{M}^o_a$, then it has to be the case that *the interval* J lies in *the interval* I.

8 $R = 5\,import$.
If the premiss $\neg\Box^o_a A$ lies in interval $I \in \mathbf{I}$ where $o \in O_{5\,import} (\subseteq Modalities)$, $a \in People$ and the conclusion $E = \neg\Box^o_a A$ lies in the interval $J \in \mathbf{M}^o_a$, then it has to be the case that *the interval* J lies in *the interval* I.

9 $R = B\,import$.
If the premiss A lies in interval $I \in \mathbf{I}$ where $o \in O_{B\,import} (\subseteq Modalities)$, $a \in People$ and the conclusion $E = \neg\Box^o_a\neg A$ lies in the interval $J \in \mathbf{M}^o_a$, then it has to be the case that *the interval* J lies in *the interval* I.

10 $R = K\,export$.
If the premiss A lies in interval $I \in \mathbf{M}^o_a$ where $o \in Modalities$, $a \in People$ and the conclusion $E = \Box^o_a A$ lies in the interval $J \in \mathbf{I}$, then it has to be the case that *the interval* I lies *in the interval* J.

11 $R = D\,export$.
If the premiss A lies in interval $I \in \mathbf{M}^o_a$ where $o \in O_{D\,export} (\subseteq Modalities)$, $a \in People$ and the conclusion $E = \neg\Box^o_a\neg A$ 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 $I \in \mathbf{M}^o_a$ where $o \in O_{T\,export} (\subseteq Modalities)$, $a \in People$ 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 demand that the agent-index and 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 O_{rule} for which the rule is to hold. For the K-rules this conditon is vacuous, $O_K = Modalities$, since these rules hold for all normal modal operators.

The combined system for knowledge and belief, where we have the logic $KT45_{(m)}$ for knowledge (operators K_a) and the logic $KD45_{(m)}$ for belief (operators B_a), can now be defined as a Fitch-style deduction system in the following way. Taking knowledge operators K_a to have operator-index 1 ($K_a = \Box^1_a$) and belief operators B_a to have index 2 ($B_a = \Box^2_a$), we define the O_{Rule} as: $O_{K\,import} = O_{K\,export} = O_{4\,import} = O_{5\,import} = \{1, 2\}$, $O_{D\,export} = \{2\}$ and $O_{T\,export} = \{1\}$ ($O_{B\,import} = \emptyset$).

4.1.2. Interacting modalities

In combining a logic for knowledge with one for belief, we have achieved a (multi-agent) multi-modal deduction system where strict subordinate proofs for different operators may be used in one proof figure. However, the modalities still lead separate lives inside the derivations each with its own strict subordinate proofs and import and export rules.

In this section we investigate whether interactions between the modalities can be expressed deductively, starting from the remaining part of the system KB (KB_{CD} without the axioms involving Common Knowledge and Common Belief) by looking at the axioms relating knowledge and belief. From there we move to more general deductive interaction patterns.

Interaction between Knowledge and Belief

By combining the machinery for reasoning about knowledge and about belief we get a system with K_a as well as B_a-subordinate proofs, where the interesting question is how these two notions interact. There are several intuitively plausible interactions between the knowledge and the beliefs of an agent. The system KB has two, formalized in the axioms $K_a\varphi \supset B_a\varphi$ ($A15$) and $B_a\varphi \supset K_aB_a\varphi$ ($A16$). Deductively these axioms correspond to variations on an import rule.

The first axiom, $K_a\varphi \supset B_a\varphi$, states 'if you know something, you believe it': since belief is less certain than knowledge, being sure enough to know something implies that you are sure enough to believe it. In model theoretical terms, every world that is still possible according to your knowledge (epistemic alternative), should also be possible according to your beliefs (boulomaic alternative). Deductively this means that every strict subordinate K_a-proof is also a strict subordinate B_a-proof, propositions that are known by some agent may be used when reasoning in a 'belief'-strict subordinate proof of this agent.

To achieve this, a kind of import rule has to 'force' the import of a proposition that is 'known' into a 'belief-strict subordinate proof':

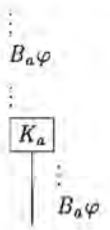
$\vdots$
$K_a\varphi$
$\vdots$
$\boxed{B_a}$
$\quad\vdots$
$\quad\varphi$

Since this rule looks very much like the K-import rule, we shall call it 'FK-import' for 'forced K-import'. Given the normality of the knowledge and belief operators, it is easy to see that the axiom can be derived by the rule and that the rule is derivable in the presence of the axiom.

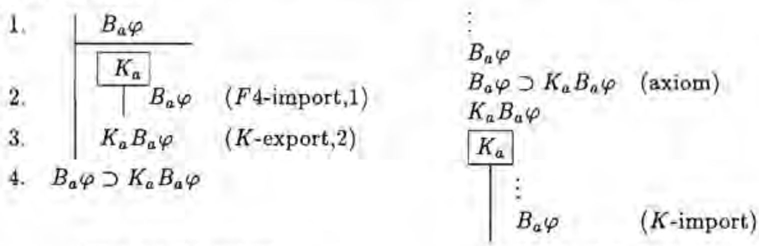
From rule to axiom		From axiom to rule	
1. $K_a\varphi$		$\vdots$	
$\boxed{B_a}$		$K_a\varphi$	
2. $\quad\varphi$	(FK-import,1)	$K_a\varphi \supset B_a\varphi$	(axiom)
3. $B_a\varphi$	(K-export,2)	$B_a\varphi$	
4. $K_a\varphi \supset B_a\varphi$		$\boxed{B_a}$	
		$\quad\vdots$	
		$\quad\varphi$	(K-import)

The axiom $B_a\varphi \supset K_aB_a\varphi$ is a further elaboration of the dependency between knowledge and belief: if you believe something, you know that you believe it. This means that beliefs are 'conscious' in the sense that an agent knows he has them. In adopting this axiom Kraus and Lehman chose a very explicit notion of belief, others argue that agent can have beliefs they are less 'aware' of, leading to various weakenings of the traditional boulomaic logic.

Model theoretically the axiom says that if an agent believes something, he will still believe it in all his epistemic alternatives. In terms of deduction, a 'belief-formula' ($B_a\varphi$) of an agent may be used when reasoning in a 'knowledge-strict subordinate proof of that agent. To achieve this an import rule has to 'force' the import of formulas of the form $B_a\varphi$ into a K_a-subordinate proof:



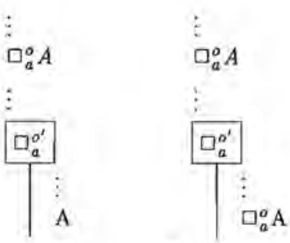
We shall call it '$F4$-import' for 'forced 4-import'. Given the normality of the knowledge and belief operators the axiom can be derived by the rule and the rule is derivable in the presence of the axiom.



From rule to axiom From axiom to rule

These rules can be formulated in a more general way in the system $\Box PROP_{fitch}^{People,Modalities}$, by taking the sets O_{FK} and O_{F4} to be elements of *Modalities* × *Modalities*. This is useful, since we will encounter other pairs of operators which are structurally related in the same way.

4.1.5. DEFINITION. **Deduction rules**
These interactions can be brought into the deduction-system by adding the following rules:



FK-import $F4$-import

4.1.6. DEFINITION. **Application of deduction rules**
Given a proof figure $\mathbf{D}$, with interval $D = [1, n]$, formulas $F_1, \ldots, F_n$ 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 $R = FK$ *import*.
If the premiss $\Box_a^o A$ lies in interval $I \in \mathbf{I}$ $a \in People$ and the conclusion $E = A$ lies in the interval $J \in \mathbf{M}_a^{o'}$ where $(o, o') \in O_{FK}(\subseteq Modalities \times Modalities)$, then it has to be the case that *the interval* J lies in *the interval* I.

14 $R = F4$ *import*.
If the premiss $\Box_a^o A$ lies in interval $I \in \mathbf{I}$ $a \in People$ and the conclusion $E = \Box_a^o A$ lies in the interval $J \in \mathbf{M}_a^{o'}$ where $(o, o') \in O_{F4}(\subseteq Modalities \times Modalities)$, then it has to be the case that *the interval* J lies in *the interval* I.

Under the indexing convention for the operators used previously, the forced import rules for knowledge and belief are the cases where $O_{FK} = \{1, 2\}$ and $O_{F4} = \{2, 1\}$.

Equivalence of K and B

Adding axioms relating knowledge and belief is not without danger, axioms that are intuitively plausible can cause the 'collapse' of knowledge and belief: $K_a\varphi \leftrightarrow B_a\varphi$ becomes a theorem of the system. An example of such a principle is $B_a\varphi \supset B_a K_a \varphi$, expressing 'believed consciousness': an agent believes that his beliefs are conscious (knowledge).

It is easy enough to find the deduction rule corresponding to this axiom ('bc-import'):

$\vdots$
$B_a\varphi$
$\vdots$
$\boxed{B_a}$
| $\vdots$
| $K_a\varphi$

with the usual relation between the axiom and the rule:

1. | $B_a\varphi$
   |——
   | $\boxed{B_a}$
2. | | $K_a\varphi$ (bc-import 1)
3. | $B_a K_a \varphi$ (K-export 2)
4. $B_a\varphi \supset B_a K_a \varphi$

From rule to axiom

$\vdots$
$B_a\varphi$
$B_a\varphi \supset B_a K_a \varphi$ (axiom)
$B_a K_a \varphi$
$\boxed{B_a}$
| $\vdots$
| $K_a\varphi$ (K-import)

From axiom to rule

However, this rule gives rise to the collapse of K_a into B_a since it allows the derivation of $B_a\varphi \supset K_a\varphi$ in combination with the KB-rules for knowledge and belief ($K_a\varphi \supset B_a\varphi$ is an axiom):

1.	$B_a\varphi$	
	$[B_a]$	
2.	$K_a\varphi$	(bc-import 1)
3.	$B_aK_a\varphi$	(K-export 2)
4.	$\neg K_a\varphi$	
	$[K_a]$	
5.	$\neg K_a\varphi$	(5-import 4)
6.	$K_a\neg K_a\varphi$	(K-export 5)
7.	$B_aK_a\varphi$	(reiteration 3)
	$[B_a]$	
8.	$\neg K_a\varphi$	(FK-import 6)
9.	$\neg B_a\neg\neg K_a\varphi$	(D-export 8)
10.	$\neg B_aK_a\varphi$	
11.	$\neg\neg K_a\varphi$	
12.	$K_a\varphi$	
13.	$B_a\varphi \supset K_a\varphi$	

Deriving $B_a\varphi \supset K_a\varphi$ depends on the following rules for K_a and B_a, besides the basic K-rules and the bc-import rule:

FK-import for (K, B), corresponding to $K_a\varphi \supset B_a\varphi$,

5-import, negative introspection for knowledge $(\neg K_a\varphi \supset K_a\neg K_a\varphi)$,

D-export, consistency of belief $(B_a\varphi \supset \neg B_a\neg\varphi)$.

This squares with the analysis of van der Hoek that one of these three principles is to be given up if the axiom $B_a\varphi \supset B_aK_a\varphi$ is to be added consistently.

Avoiding the collapse of the knowledge into belief is not a trivial matter. In [Van der Hoek 1992] it is shown that system KB is *saturated* in the sense that adding any axiom (of a certain general form) relating knowledge and belief will cause a collapse. We shall not repeat his discussion for the natural deduction case, but rather look at the general forms for interaction axioms resulting from his investigations to see if they have deductive counterparts.

General interaction patterns

Earlier we gave a general formulation of the FK-rule for pairs of modal operators, allowing for the derivation of the axiom $\Box_a^o\varphi \supset \Box_a^{o'}\varphi$ for any pair $(o, o') \in O_{FK}$. If we concentrate on the modalities by using $X, Y, \ldots$ for the different epstemic operators (and forgetting about the agent index for the moment), this yields interaction axioms of the form $X\varphi \supset Y\varphi$. Axioms of this form express an 'ordering' of two modalities, X is 'stronger' than Y: whenever something is an 'X-strong' belief it is also a 'Y-strong' belief.

In the same way the $F4$-import rule corresponds to axioms of the form $X\varphi \supset YX\varphi$. This turns out to be an instantiation of one of the following general forms of interactions between modal operators defined in [Van der Hoek 1992]. Let X, Y, Z range over epistemic operators. Then, formulas of the form

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

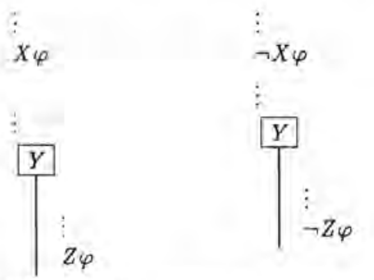
c) $XY\varphi \supset Z\varphi$ are called positive extraspection (pe. -) formulas

d) $X\neg Y\varphi \supset \neg Z\varphi$ are called negative extraspection (ne. -) formulas

e) $X(Y\varphi \supset \varphi)$ are called trust formulas

Instantiations of a) - d) are collectively referred to as *inspection formulas.*

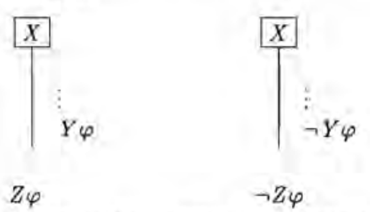
Given only that X, Y, and Z are normal operators, the introspection formulas correspond to import rules and the extraspection formulas to export rules:



positive introspection negative introspection

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 negative extraspection

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.

We have already encountered cases of introspection: the 4 and 5-import rules are the simplest possible cases of positive and negative introspection ($X = Y = Z$) and $F4$-import is also an instance of positive introspection ($X = Z$). An example of positive extraspection is $KB\varphi \supset B\varphi$ ($Y = Z$), which holds for all agents in KB. However, since this already follows by the reflexivity of the K-operator (T-export), the corresponding rule is not needed.

The general reading of trust formulas $X(Y\varphi \supset \varphi)$ is that an agent believes with X-certainty that his Y-believes are true. An instance of this is $B_a(B_a\varphi \supset \varphi)$ which is theorem of KB, showing that agents in this system strongly trust their beliefs. It is difficult to find a general deduction rule for trust formulas $X(Y\varphi \supset \varphi)$. They are of course trivially derivable if the Y-modality is reflexive (e. g. $B(K\varphi \supset \varphi)$), but this condition need not hold since we only know that Y is normal. Therefore a general deduction rule has to ensure that *φ is derivable from $Y\varphi$ inside X-subordinate proofs*. This dependency does not fit the format of the rules for inspection formulas. However given certain inspection formulas for a pair of modalities the trust formula for these modalities is derivable.

4.1.7. PROPOSITION. *A trust formula $X(Y\varphi \supset \varphi)$ is derivable in $\Box PROP_{fitch}^{People,Modalitities}$ for normal modal operators X and Y, if $\Box PROP_{fitch}^{People,Modalitities}$ has FK-import and $F5$-import for (X, Y) (negative introspection where $X = Z$).*

PROOF.

1.	$Y\varphi \vee \neg Y\varphi$	(propositional logic)
2.	│ $Y\varphi$	
	│ [X]	
3.	│ │ φ	(FK-import 2)
4.	│ │ $\neg Y\varphi \vee \varphi$	($\vee$-intro 3)
5.	│ │ $Y\varphi \supset \varphi$	(prop log 4)
6.	│ $X(Y\varphi \supset \varphi)$	(K-export 5)
7.	│ $\neg Y\varphi$	
	│ [X]	
8.	│ │ $\neg Y\varphi$	($F5$-import 7)
9.	│ │ $\neg Y\varphi \vee \varphi$	($\vee$-intro 8)
10.	│ │ $Y\varphi \supset \varphi$	(prop log 9)
11.	│ $X(Y\varphi \supset \varphi)$	(K-export 10)
12.	$X(Y\varphi \supset \varphi)$	($\vee$-elim 1,6,11)

In the above we have read X, Y, and Z as different modal operators, tacitly assuming that they had the same agent index. They could equally well be read as expressing dependencies between different 'copies' of the same operator and hence as principles for the interactions between (beliefs of) agents. Under this interpretation the FK-rule corresponds to axioms like $K_1\varphi \supset K_2\varphi$, ordering agents according to their information states: agent 2 knows (at least) everything that agent 1 knows.

Hence in their most general form interaction rules could vary both operator and agent indices. In the format of $\Box PROP_{fitch}^{People,Modalities}$ this would mean stating the rules with respect to *two* sets:

$O_{rule} \in$ *Modalities* $\times$ *Modalities*

$A_{rule} \in$ *People* $\times$ *People*

In this way higly specific interactions can be expressed like $\neg B_1\varphi \supset K_2\neg K_3\varphi$: 'If it is not the case that agent 1 believes that φ, then agent 2 knows that agent 3 does not know φ.

4.1.3. Group modalities

So far we have been looking at multi-agent logic as an aggregation of one-agent epistemic logics: each agent has his own operators for knowledge and belief, but there are modalities which depend on more than one agent. In this section we investigate notions of knowledge and belief that belong to a *group* of agents.

Everybody knows and Common Knowledge

A very simple group modality is 'Everybody knows' that φ, which can be formalized by the definable operator 'E', where $E\varphi = K_1\varphi \wedge \ldots \wedge K_m\varphi$ ($People = \{1, \ldots, m\}$). Using this operator we can express the much more complex notion of 'Common Knowledge'. Something is common knowledge of a group of agents if it is not only known by all members of the group, but it is also known by all members that it is known by all members, which in turn is also known by all members ... and so on. Hence $C\varphi =_{def} E\varphi \wedge EE\varphi \wedge \ldots$, an infinite conjunction.

Still, it can be axiomatized and these axioms have familiar forms. That C is a normal operator is expressed by $A4)$ $C(\varphi \supset \psi) \supset (C\varphi \supset C\psi)$ (and $R1$ $\vdash \varphi \Rightarrow \vdash C\varphi$), hence we have the K-rules for C. The axioms relating C and E also look familiar: $A5)$ $C\varphi \supset E\varphi$ and $A6)$ $C\varphi \supset EC\varphi$. They correspond to the FK- and $F4$-import rule for (C, E).

However, since E is a definable modality we have not yet added it to our deduction system, there are no E-strict subordinate proofs or import and export rules for this operator. We don't need them: adopting the FK- and $F4$-import rules for the pair (C, K_a) (corresponding to the weaker axioms $C\varphi \supset K_a\varphi$ and $C\varphi \supset K_aC\varphi$) is sufficient. The axioms relating Common Knowledge and Everybody's Knowledge now follow from the rules for C and '$\wedge$', and the definition of E:

$C\varphi$	$C\varphi$
K_1	K_1
φ (FK-import)	$C\varphi$ ($F4$-import)
$K_1\varphi$ (K-export)	$K_1 C\varphi$ (K-export)
⋮	⋮
K_n	K_n
φ (K-import)	$C\varphi$ (FK-import)
$K_n\varphi$	$K_n C\varphi$
$K_1\varphi \wedge K_2\varphi$ ($\wedge$-intro)	$K_1 C\varphi \wedge K_2 C\varphi$ ($\wedge$-intro)
⋮	⋮
$K_1\varphi \wedge \ldots \wedge K_n\varphi$	$K_1 C\varphi \wedge \ldots \wedge K_n C\varphi$
$E\varphi$ (definition)	$EC\varphi$ (definition)
$C\varphi \supset E\varphi$	$C\varphi \supset EC\varphi$

(Vertical dots indicate a 'repetition of moves' for other agent indices.)

4.1.8. EXAMPLE. Given the rules relating C and K_a it is also easier to see how, for a given proposition φ, any (finite) prefix of operators $K_a, \ldots, K_{a'}$ can be derived from $C\varphi$.

1.	$C\varphi$	
	K_2	
2.	$C\varphi$	($F4$-import 1)
	K_5	
3.	$C\varphi$	($F4$-import 2)
	K_2	
4.	$C\varphi$	($F4$-import 3)
	K_3	
5.	$C\varphi$	($F4$-import 4)
	K_3	
6.	φ	(FK-import 5)
7.	φ	(T-export 6)
8.	$K_3\varphi$	(K-export 7)
9.	$K_2K_3\varphi$	(K-export 8)
10.	$K_5K_2K_3\varphi$	(K-export 9)
11.	$K_2K_5K_2K_3\varphi$	(K-export 10)
12.	$C\varphi \supset K_2K_5K_2K_3\varphi$	

The strategy is to set up a chain of strict subordinate proofs indexed $K_a, \ldots, K_{a'}$ and using the

$F4$-import rule to propagate the formula $C\varphi$. From $C\varphi$ in the $K_{a'}$-subordinate proof a final $(K_{a'})$-subordinate proof is started, importing $C\varphi$ as φ using the FK-import rule. Ordinary T-export then yields φ in the original $K_{a'}$ subordinate proof. Now all subordinate proofs can be closed successively by means of K-export, yielding the formula $K_a \ldots K_{a'}\varphi$. Hence a satisfactory deductive account of the elimination of the C can be given. Discussion of the introduction of the C operator as recorded in the so-called 'induction' axiom $A7$) $C(\varphi \supset E\varphi) \supset (\varphi \supset C\varphi)$ is postponed until we can treat it jointly with its belief counterpart $A13$.

Common Belief and Everybody believes

In analogy to the group modalities for knowledge we have the notions 'Everyone believes', $F\varphi = B_1\varphi \wedge \ldots \wedge B_m\varphi$ ($People = \{1, \ldots, m\}$) and 'Common Belief' $D\varphi = F\varphi \wedge FF\varphi \wedge \ldots$. Since the first three axioms for D and F exactly match those for C and E, we have K-import and export for D ($A10$) and FK- and $F4$-import for the pair (D, B_a) $((A11), (A12))$.

The difference in strength between the two 'common operators' is expressed in the axiom $A16$) $C\varphi \supset D\varphi$, if something is Common Knowledge it is also Common Belief, which corresponds to an instantiation of the FK-import rule for (C, D).

Up to this point it has been straightforward to give a Fitch-style deduction account of the system KB_{CD} using only generalized versions of our original monological rules and two new interaction rules. Unfortunately this format does not extend to cover the two remaining axioms of KB_{CD}. Just like Common Knowledge, Common Belief is introduced by means of an induction axiom $A13$) $D(\varphi \supset F\varphi) \supset (F\varphi \supset D\varphi)$. It is an open question whether there is a deductive counterpart of this axiom (or $A7$) $C(\varphi \supset E\varphi) \supset (\varphi \supset C\varphi)$). However, the situation can be saved ungracefully by simply adding the 'induction axioms' to $\Box PROP_{fitch}^{People, Modalities}$ *as axioms.*

Implicit Knowledge

Not all group modalities refer to situations where knowledge or belief of a proposition is shared by all members of the group. An interesting modality contrasting with Common Knowledge and Everbody's Knowledge in this respect is 'implicit knowledge', the knowledge that is implicitly available within a group: everything that can be derived from the combined knowledge of the agents.

This kind of knowledge will be denoted by the operator 'I', 'it is implicitly known that ...', which is characterized by the following rule:

$$\textit{Implicit Normality} \quad \frac{(\varphi_1 \wedge \varphi_2 \ldots \wedge \varphi_n) \supset \psi}{(K_a\varphi_1 \wedge \ldots \wedge K_{a'}\varphi_n) \supset I\psi} \; (a, \ldots, a' \in People)$$

The rule shows that in addition to the knowledge of each of the individual agents (taking all knowledge operators to have the same agent-index), implicit knowledge also consists of knowledge 'distributed' over a number of agents, e. g. :

$$\frac{(\varphi \wedge (\varphi \supset \psi)) \supset \psi}{(K_1\varphi \wedge K_2(\varphi \supset \psi)) \supset I\psi}$$

If agent 1 knows $\varphi \supset \psi$ and agent 2 knows φ then together they implicitly know ψ.

Implicit knowledge is of interest in connection with information dialogues: if we think of the dialog participants as agents with information states represented by epistemic formulas, then implicit knowledge precisely defines the propositions the participants could conclude to during an information dialogue:

> The propositions that are known by one (or more) of the agents before the beginning of the dialogue.
>
> Propositions that are not known to any participant before the dialogue, but that follow from combining information of individual agents

An example of the latter would be a meeting of three mathematicians at a conference, where each has proved a different lemma and the combination of the lemmas proves Goldbach's Conjecture. After the conversation each of the mathematicians will know the proof of the (ex-) Conjecture, whereas none of them knew it before[1]

Clearly the I-operator is a normal operator, so we start its deductive characterization by introducing I-strict subordinate proofs and adopting K-rules for it. To obtain *implicit normality*, we add FK-import for the pair (K_a, I):

$$\begin{array}{l} \vdots \\ K_a\varphi \\ \boxed{I} \\ \quad\Big| \quad \vdots \\ \quad\Big| \quad \varphi \end{array}$$

Using this rule formulas of the form $K_a\varphi$ occuring in the main proof may be reiterated as φ in the I-strict subordinate proof, *regardless of the agent index a*. In this way propositions known by different agents can be imported and combined in a single subordinate proof:

1.	$K_1\varphi \wedge K_2(\varphi \supset \psi)$	
2.	$K_1\varphi$	
3.	$K_2(\varphi \supset \psi)$	
	$\boxed{I}$	
4.	$\quad \varphi$	(FK-import 2)
5.	$\quad \varphi \supset \psi$	(FK-import 3)
6.	$\quad \psi$	
7.	$I\psi$	

Implicit knowledge is of model theoretic interest as noted in ([Van der Hoek 1992]) because interpreting it in the most straightforward way, with R_I ranging over the worlds in the intersection of $R_1, \ldots R_n$, leads to peculiarities. The Kripke models obtained in this way are

[1] J.J.Ch.Meyer gives this example for Fermat's Last Theorem ([Meyer 1994]), but we have modified it slightly in the light of recent developments in mathematics.

sound and complete with respect to the *implicit normality*-rule, but also with respect to the epistemic logics extended with the axiom(schema)

$$K_a\varphi \supset I\varphi \qquad a = \{1, \ldots, m\}.$$

Clearly the logic with the axiom is weaker than the logic with the rule, since the former rules out that implicit knowledge could be derived from the knowledge of a group of agents. For the logic with the axiom we have $K_1\varphi \vee \ldots \vee K_m\varphi \leftrightarrow I\varphi$, for the logic with the rule only half of this equivalence holds: $K_1\varphi \vee \ldots \vee K_m\varphi \supset I\varphi$.

Deductively the 'weak' implicit knowledge corresponding to the axiomschema puts a restriction on the FK-import of K_a-formulas into I-strict subordinate proofs: all knowledge formulas that are imported have to be of the same agent index. Since this applies for each agent, this means operationally that once a strict I-proof is started from a formula $K_a\varphi$ only formulas of agent-index a may be imported in this I-strict proof. The conclusions $I\psi$ of such subordinate proofs will be known by the particular agent a (they are consequences of his knowledge) and hence $I\varphi \supset K_1\varphi \vee \ldots \vee K_m\varphi$ is satisfied.

Although this restricition, which still needs to be formulated formally, prevents the direct derivation of 'distributed conclusions', these can still be derived indirectly:

1.	$K_1(\varphi \supset \psi) \wedge K_2\varphi$	
2.	$K_1(\varphi \supset \psi)$	
3.	$K_2\varphi$	
	$\boxed{I}$	
4.	$\quad \varphi \supset \psi$	(FK-import for agent-index 1)
5.	$I(\varphi \supset \psi)$	
	$\boxed{I}$	
6.	$\quad \varphi$	(FK-import for agent-index 2)
7.	$I\varphi$	
	$\boxed{I}$	
8.	$\quad \varphi \supset \psi$	(K-import 5)
9.	$\quad \varphi$	(K-import 7)
10.	$\quad \psi$	
11.	$I\psi$	
12.	$(K_1(\varphi \supset \psi) \wedge K_2\varphi) \supset I\psi$	

The idea is to derive distributed knowledge by first turning the premisses $(K_a\varphi_1, \ldots, K_{a'}\varphi_n)$ known by the agents involved into implicit knowledge $(I\varphi_1, \ldots, I\varphi_n)$. This is done separately for each agent, using the FK-rule with the condition. After that, we use the normality of I to start an additional subordinate I-proof, into which each of these I-formulas is imported using K-import. In this way the desired 'distributed conclusion' $(I\psi)$ can be obtained after all.

Hence a weak implicit knowledge operator cannot be normal, it has only the K-export rule, combined with the conditional $F4$-rule. The problem of formulating the condition in the framework of $\Box PROP_{fitch}^{People,Modalities}$ remains. Since all the import rules are defined for individual formulas, there does not seem to be a natural way to ensure that once a I-strict subordinate proof is started from a formula with operator K_a, all other imported formulas are of that index.

There is a way of formulating Fitch-style modal deduction system, that does allow for a natural expression of the conditional $F4$ rule. In [Wansing 1995], Heinrich Wansing gives an inductive definition of the Fitch-style modal deduction system. Expressions are of the form $\mathcal{P}_N(|\Pi,\ A,\ \Delta)$, to be read as '$\Pi$ is a proof in natural deduction of A from the finite set of assumptions Δ. A natural deduction proof (figure) can then be build up starting from the basic clause $\mathcal{P}_N(|A, A, \{A\})$, using rules like ($\supset$-elim):

$\mathcal{P}_N(|\Pi_1, A, \Delta_1), \mathcal{P}_N(|\Pi_2, A \supset B, \Delta_2) \ \Rightarrow$

$$\mathcal{P}_N(\ \left| \begin{array}{l} \Pi_1 \\ \Pi_2 \\ B \end{array} \right. \quad B,\ \Delta_1 \cup \Delta_2 \ )$$

The basic modal rule is

$\mathcal{P}_N(|\Pi,\ A,\ \Delta) \Rightarrow$

$$\mathcal{P}_N(\ \left| \begin{array}{l} \boxed{\Box}\ \Pi \\ \Box A \end{array} \right. \quad \Box A,\ \Box\Delta \ )$$

acting as K-import and K-export at once, this rule allows you to turn a propositional deduction proof Π of A from hypotheses Δ into a strict subordinate proof of $\Box A$ from $\Box\Delta$. The important difference is that the assumptions in Δ are treated *as a set*. Instead of individually importing each of the modal assumptions, the rule simultaneously prefixes all assumptions in Δ with a $\Box$ ($\Delta \Rightarrow \Box\Delta$).

In this setting the weak implicit knowledge operator corresponds naturally to the following rule:

$\mathcal{P}_N(|\Pi,\ A,\ \Delta) \Rightarrow$

$$\mathcal{P}_N(\ \left| \begin{array}{l} \boxed{I}\ \Pi \\ IA \end{array} \right. \quad IA,\ K_a\Delta \ )$$

A proof Π of A from Δ, may be turned into an I-strict subordinate proof yielding IA from $K_a\Delta$ for some agent index a. Since the assumptions of the proof are modalized *as a set*, all assumptions will be propositions known by one particular agent a.

In the inductive system it would be difficult to formulate a rule expressing directly the derivation of a distributed conclusion, but the strong notion of implicit knowledge can be obtained by adding the normality rule for I-proofs:

$\mathcal{P}_N(|\Pi,\ A,\ \Delta) \Rightarrow$

$$\mathcal{P}_N(\ \left| \begin{array}{l} \boxed{I}\ \Pi \\ IA \end{array} \right. \quad IA,\ I\Delta \ )$$

Distributed conclusions are then derivable in the indirect way described earlier.

4.2. Type theory

Based on the earlier work for monological modal system we can straightforwardly transfer the multi-agent and multi-modal extensions of the Fitch-style natural deduction to the MPTS's. When that has been done, we review the modal deduction rules defined earlier from the perspective of the multi-agent multi-modal system.

4.2.1. MPTS with multiple agents and modalities

In analogy with $\Box PROP_{fitch}^{People,Modalities}$ we generalize the MPTS's to the multi-agent multi-modal case by 'indexing everything' with respect to the set of agents *People*, and the set of operators *Modalities*: modal operators ($\Box_a^o$), generalized contexts ($\boxdot_a^o$) and proof functions ($\check{k}_a^o, \hat{k}_a^o, \ldots$). Here are the definitions for $MPTS_\beta^{People,Modalities}$:

4.2.1. Definition. **Pseudoterms**
The set of pseudoterms $\mathcal{T}$ over $\mathcal{S}$, *People* (for all $a \in People$) and *Modalities* (for all $o \in Modalities$) is:

$$\mathcal{T} ::= \mathcal{S} | Var | \Pi\, VAR : \mathcal{T}.\mathcal{T}) | (\lambda\, Var : \mathcal{T}.\mathcal{T}) | \mathcal{T}\mathcal{T} | \Box_i^j \mathcal{T} | \check{k}_i^j \mathcal{T} | \check{4}_a^o \mathcal{T} | \check{5}_a^o \mathcal{T} | \check{b}_a^o \mathcal{T} | \hat{k}_i^j \mathcal{T} | \hat{d}_a^o \mathcal{T} | \hat{t}_a^o \mathcal{T}$$

4.2.2. Definition. **Contexts**
In generalized contexts each separator is now indexed with an $a \in People$ and an $o \in Modalities$:

(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: $x_1 : A_1, \ldots, x_n : A_n$.

(iii) A *generalized pseudo-context* is a finite ordered sequence of pseudo-contexts and indexed separators: $G = \Gamma_1 \boxdot_a^o \ldots \boxdot_{a'}^{o'} \Gamma_n$ with $a, \ldots, a' \in People$, $o, \ldots, o' \in Modalities$.

4.2.3. Definition. **Multi-agent multi-modal Modal Pure Type Systems**
A *multi-agent multi-modal Pure Type System with β-conversion*, $\mathrm{MPTS}_\beta^{People,Modalities}$, is given by a set $\mathcal{S}$ of *sorts* containing *Prop*, *Set*, and *Type*, a set $\mathcal{A}^{Type} \subset \mathcal{S} \times \mathcal{S}$ of *typing axioms*, a set $\mathcal{A}^{Logic} \subset \mathcal{C} \times \mathcal{T}$ 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 $\Box\lambda_\beta(\mathcal{S}, \mathcal{A}, \mathcal{R})$ and is the typed λ-calculus with the following deduction rules:

$$(axioms)\quad \varepsilon \vdash s_1 : s_2 \quad \text{if } s_1 : s_2 \in \mathcal{A}^{Type} \qquad \varepsilon \vdash c : A : Prop \quad \text{if } c : A \in \mathcal{A}^{Logic}$$

$$(start)\quad \frac{G \vdash A : s}{G,\ x : A \vdash x : A}$$

$$(weakening)\ \frac{G \vdash A : B \quad G \vdash C : s}{G,\ x : C \vdash A : B}$$

$$(product)\ \frac{G \vdash A : s_1 \quad G,\ x : A \vdash B : s_2}{G \vdash (\Pi x : A.B) : s_3} \quad if\ (s_1, s_2, s_3) \in \mathcal{R}$$

$$(application)\ \frac{G \vdash F : (\Pi x : A.B) \quad G \vdash a : A}{G \vdash Fa : B[x := a]}$$

$$(abstraction)\ \frac{G,\ x : A \vdash b : B \quad G \vdash (\Pi x : A.B) : s}{G \vdash (\lambda x : A.b) : (\Pi x : A.B)}$$

$$(conversion)\ \frac{G \vdash A : B \quad G \vdash B' : s \quad B =_\beta B'}{G \vdash A : B'}$$

$$(boxing)\ \frac{G \vdash A : Prop}{G \vdash \Box^o_a A : Prop} \quad (\forall o \in Modalities,\ \forall a \in People)$$

$$(transfer_1)\ \frac{G \vdash A : s}{G \boxdot^o_a \varepsilon \vdash A : s} \quad (\forall o \in Modalities, \forall a \in People)$$

$$(transfer_2)\ \frac{G \vdash A : B : \Box}{G \boxdot^o_a \varepsilon \vdash A : B} \quad (\forall o \in Modalities, \forall a \in People)$$

$$(transfer_3)\ \frac{G \vdash A : B : Set}{G \boxdot^o_a \varepsilon \vdash A : B} \quad (\forall o \in Modalities, \forall a \in People)$$

$$(transfer_{ax})\ \frac{G \vdash c : A : Prop}{G \boxdot^o_a \varepsilon \vdash c : A} \quad (\forall o \in Modalities, \forall a \in People,\ c : A \in \mathcal{A}^{Logic})$$

$$(K\ import)\ \frac{G \vdash A : \Box^o_a B : Prop}{G \boxdot^o_a \varepsilon \vdash \check{k}^o_a A : B} \quad (\forall o \in Modalities, \forall a \in People)$$

$$(K\ export)\ \frac{G \boxdot^o_a \varepsilon \vdash A : B : Prop}{G \vdash \hat{k}^o_a A : \Box^o_a B} \quad (\forall o \in Modalities, \forall a \in People)$$

$$(4\ import)\ \frac{G \vdash A : \Box^o_a B : Prop}{G \boxdot^o_a \varepsilon \vdash \check{4}^o_a A : \Box^o_a B} \quad (\forall o \in O_{4\ import}(\subseteq Modalities), \forall a \in People)$$

$$(5\ import)\ \frac{G \vdash A : \neg\Box^o_a B : Prop}{G \boxdot^o_a \varepsilon \vdash \check{5}^o_a A : \neg\Box^o_a B} \quad (\forall o \in O_{5\ import}(\subseteq Modalities), \forall a \in People)$$

$$(B\ import)\ \frac{G \vdash A : B : Prop}{G \boxdot^o_a \varepsilon \vdash \check{b}^o_a A : \neg\Box^o_a \neg B} \quad (\forall o \in O_{B\ import}(\subseteq Modalities), \forall a \in People)$$

$$(D\ export)\ \frac{G \boxdot^o_a \varepsilon \vdash A : B : Prop}{G \vdash \hat{d}^o_a A : \neg\Box^o_a \neg B} \quad (\forall o \in O_{D\ export}(\subseteq Modalities), \forall a \in People)$$

$$(T\ export)\ \frac{G\ \boxdot^o_a\ \varepsilon\ \vdash\ A : B : Prop}{G\ \vdash\ \hat{\imath}^o_a A : B}\quad (\forall o \in O_{T\ export}(\subseteq Modalities), \forall a \in People)$$

s ranges over the S the set of sorts, x ranges over variables, c over constants, o ranges over the set *Modalities* of operator indices, a ranges over the set *People* of agent indices, and it is assumed that in the rules (*start*) and (*weakening*), the newly declared variable is always fresh.

Compared with the definition of $MPTS_\beta$ the non-modal rules upto Conversion are unchanged. The Boxing rule and the Transfer rules now have quantification over all agent indices and all modal operators. Hence Boxing enables us to modalize a proposition with any modal operator $\Box^o_a$ allowed by *People* and *Modalities*. For the Transfer rules the quantification means that non-proof statements and (logical) axioms may be transferred into any subcontext, over every separator $\boxdot^a_o$ regardless of the operator- and agent-index.

Since the K-rules are supposed to hold for all normal modal operators, they only require that the indices o and a of the modal operator, the subordinate context, and the proof function match. The other import and export rules may not hold for all operators in *Modalities*, hence they are parametrized with respect to a subset O_{rule} of *Modalities*.

Given the correspondence between modal natural deduction and MPTS's discussed earlier, it is evident that we have to index the separators '$\boxdot$', in the generalized contexts. These indicate the subordinate contexts that are the analogons of the strict subordinate proofs in modal natural deduction. Hence if we discern subordinate proofs for different agents and modalities in natural deduction we should do likewise with subordinate contexts in type theory by indexing the separators.

The indexing of the proof functions ($\check{k}^a_o, \ldots$) deserves some discussion. Let's start with the agent-index by looking at the logic $KT45_m$ for knowledge as an instantiation of the general system above (writing K_a in stead of $\Box^1_a$):

$$(K\ import)\ \frac{G\ \vdash\ M : K_a A : Prop}{G\ \boxdot_a\ \varepsilon\ \vdash\ \check{k}_a M : A}\qquad (K\ export)\ \frac{G\ \boxdot_a\ \varepsilon\ \vdash\ M : A : Prop}{G\ \vdash\ \hat{k}_a M : K_a A}$$

$$(4\ import)\ \frac{G\ \vdash\ M : K_a A : Prop}{G\ \boxdot_a\ \varepsilon\ \vdash\ \check{4}_a M : K_a A}\qquad (5\ import)\ \frac{G\ \vdash\ M : \neg K_a A : Prop}{G\ \boxdot_a\ \varepsilon\ \vdash\ \check{5}_a M : \neg K_a A}$$

$$(T\ export)\ \frac{G\ \boxdot_a \varepsilon\ \vdash\ M : A : Prop}{G\ \vdash\ \hat{\imath}_a M : A}$$

In indexing the proof functions with respect to the agents, we emphasize the structural similarity between the modal rules for the different copies of the knowledge operator.

In terms of Kripke-semantics, these functions change the proofs of epistemic formulas an agent has in the current world into proofs of a related formula in that agents epistemic alternatives and vice versa. Viewed in this way, using indexed versions of the proof-object transformers $\check{k}, \hat{k}$ for the K_a-operators corresponds to the idea that *all agents adress their epistemic alternatives in the same way* (although epistemic alternatives may differ in content across agents). Having indexed versions of $\check{k}$ and $\hat{k}$ is in step with the 'propositions-as-types' ideology: we retain the property that the entire proof can be reconstructed from the term, without the indices we would have to refer to the type or the context during reconstruction.

Along the same lines as for knowledge we can formulate a $KD45_m$ system for belief. But if we want to use the structural similarity between the rules for knowledge and the rules for belief, as we did earlier in $\Box PROP_{fitch}^{People,Modalities}$, we have to index the separators with respect to the modal operator allowing us to discern between 'belief'-subordinate contexts and knowledge'-subordinate contexts:

$$(K\ import)\ \frac{G \vdash M : B_a A : Prop}{G \boxdot_a^B \varepsilon \vdash \check{k}_a^B M : A} \qquad (K\ export)\ \frac{G \boxdot_a^B \varepsilon \vdash M : A : Prop}{G \vdash \hat{k}_a^B M : B_a A}$$

$$(4\ import)\ \frac{G \vdash M : B_a A : Prop}{G \boxdot_a^B \varepsilon \vdash \check{4}_a^B M : B_a A} \qquad (5\ import)\ \frac{G \vdash M : \neg B_a A : Prop}{G \boxdot_a^B \varepsilon \vdash \check{5}_a^B M : \neg B_a A}$$

$$(D\ export)\ \frac{G \boxdot_a^B \varepsilon \vdash M : A : Prop}{G \vdash \hat{d}_a^B M : \neg B_a \neg A}$$

This holds for the proof functions as well: $\check{k}_a M : A$ indicates that a proof of $\Box_a$ of agent a has been changed into a proof of A in an alternative of agent a. But this holds for the 'knowledge' (K_a)- as well as the 'belief (B_a)'-interpretation of the box. We do not want to confuse the epistemic alternatives of agents a with his boulomaic alternatives: all the worlds possible according to his knowledge with all the worlds possible according to his beliefs. Hence indexing the proof functions for modal operators corresponds to the idea that *an agent accesses alternatives related to different modalities in different ways.*

For the type theoretical properties one is traditionally interested in, the generalization to more agents and modalities makes no difference.

4.2.4. THEOREM. *For λ_S a $MPTS_\beta^{People,Modalities}$ from the Modal Logic Cube, we have:*

λ_S has Unicity of Types

λ_S is Strongly Normalizing

λ_S has the Church Rosser property

λ_S enjoys Subject Reduction

PROOF. By redoing the proofs of these properties from chapter 3. This is straightforward, the agent- and operator-index do not in any way interfere with the proofs given before.

4.2.2. Interacting Modalities

The typetheoretical version of the rules relating knowledge and belief of an agent should allow the import of statements of type $K_a\varphi$ into B_a-subordinate contexts, and the import of statements of type $B_a\varphi$ in K_a-subordinate contexts respectively.

$$(FK\ import)\ \frac{G \vdash M : K_a A : Prop}{G \boxdot_a^B \varepsilon \vdash \check{f}_a^{(K,B)} M : A}$$

$$(F4\ import)\ \frac{G \vdash M : B_a A : Prop}{G \boxdot_a^K \varepsilon\ \check{g}_a^{(B,K)} M : B_a A}$$

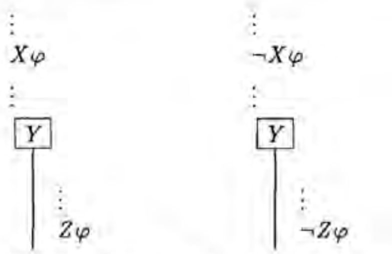
The only aspect of these rules that can not be settled by looking at their natural deduction counterparts is the treatment of the proof objects. Although we named these rules on the basis of their similarity to the K- and 4-import rules, they do present structurally new interactions: the modal operator of the formula that is to be imported no longer has to match the operator index of the strict subordinate proof.

We therefore propose to adopt new import proof functions for these rules, $\breve{f}$ for FK-import and $\breve{g}$ for $F4$-import. They are indexed with respect to agents (a) and a pair of modalities. The first operator index records the modality of the formula that was imported. In this way we remember where a formula originally came from, for instance that a formula in a B_a-subordinate proof was originally a K_a-formula (FK-import). This is trivial in a system with only two modalities, but can be necessary in systems with more modalities. The same considerations about generality lead to the recording of the modality in the subordinate proof in the second operator-index. From terms representing modal proofs in KB we would be able to infer this modality by means of the export step used to close the K_a- or B_a-subordinate contexts. However, in general a system could have rules for extraspection as well as introspection and then we may need the second operator-index to reconstruct the proof from the term. Therefore we add the interaction rules to the type theoretical system in the following general form:

$$(FK\ import)\ \frac{G\ \vdash\ M : \Box^a_o A : Prop}{G\ \boxdot^{o'}_a\ \varepsilon\ \vdash\ \breve{f}^{(o,o')}_a M : A}\quad \forall(o,o') \in O_{FK}(\subseteq Modalities), \forall a \in People$$

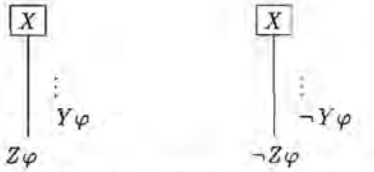
$$(F4\ import)\ \frac{G\ \vdash\ M : \Box^a_o A : Prop}{G\ \boxdot^{o'}_a\ \varepsilon\ \vdash\ \breve{g}^{(o,o')}_a M : \Box^o_a A}\quad \forall(o,o') \in O_{FK}(\subseteq Modalities), \forall a \in People$$

Since all desirable type theoretical properties are preserved under the addition of these two interaction rules, we now have the equipment to express (the rest of) the system KD_{CD} as an $MPTS_\beta^{People, Modalities}$. Before moving to the group modalities we return to the general interaction forms discussed earlier.



positive introspection   negative introspection

$$pos\ introspection\ \frac{G\ \vdash\ M : X\varphi : Prop}{G\ \boxdot^{Y}\ \varepsilon\ \vdash \breve{p}^{(X,Y,Z)} : Z\varphi}\qquad neg\ introspection\ \frac{G\ \vdash\ M : \neg X\varphi : Prop}{G\ \boxdot^{Y}\ \varepsilon\ \vdash \breve{n}^{(X,Y,Z)} : \neg Z\varphi}$$



positive extraspection   negative extraspection

$$pos\ extraspection\ \frac{G \boxdot^X \varepsilon \vdash M : Y\varphi : Prop}{G \vdash \hat{p}^{(X,Y,Z)}M : Z\varphi} \qquad neg\ extraspection\ \frac{G \boxdot^X \varepsilon \vdash M : \neg Y\varphi : Prop}{G \vdash \hat{n}^{(X,Y,Z)}M : \neg Z\varphi}$$

For these general forms we have to specify triples of modalities, the inspection rules hold for all $(o, o', o'') \in O_{rule} \subseteq (Modalities \times Modalities \times Modalities)$, for all $a \in People$. In order to record the complete proof in the term, the proof functions $(\check{p}, \hat{p}, \check{n}, \hat{n})$ are indexed with these triples (X, Y, Z). Note that the full triple is not always needed, the $F4$-rule defined above is an instance of positive introspection where $(X = Z)$.

4.2.3. Group modalities

Group modalities can now be handled in the same way as in the natural deduction system. The E and F operators are again definable (using the second order definition of '$\wedge$') and so we have the following rules for Common Knowledge and Common Belief:

The K-import and export rule for C and D

The FK- and $F4$-import rules for (C, K_a)

The FK- and $F4$-import rules for (D, B_a)

The FK-import rule for (C, D)

In the previous section we saw that there are no deductive counterparts to the induction axioms in our Fitch-style framework. Give the greater computational potential of typed λ-calculus, one could hope that some kind of 'recursive' characterization of C and D is possible. However, preliminary investigations using the powerful extension of (M)PTSs with so-called inductive types (see [Pfennig and Paulin 1990]) did not result in a (co-)inductive characterization of these operators. Hence we adopt the solution given earlier to add the induction axioms *as axioms*. Taking two fresh constants from C, we add $c_2 : \forall\alpha : Prop.(C(\alpha \supset E\alpha) \supset (\alpha \supset C\alpha))$ and $c_3 : \forall\alpha : Prop.(D(\alpha \supset F\alpha) \supset (F\alpha \supset D\alpha))$ to $\mathcal{A}^{Logic}$ as done earlier for the double negation axiom.

For (strong) implicit knowledge we have the K-rules for the I-operator and the FK-import rule for (K_a, I):

$$(FK\ import)\ \ \frac{G \vdash M : K_a A : Prop}{G \boxdot^I \varepsilon \vdash \check{f}_a^{K,I} M : A}$$

4.2.4. Modal reductions

In this section we investigate modal reduction in the framework of the multi-agent multi-modal system. We start from the monological annihilations and distributions defined earlier, and then look for new reduction possibilities.

By indexing the proof functions, the possibility to reconstruct the complete proof of a type from its term was preserved. Therefore a natural generalization of the monological modal reduction rules would take the indices into account. When the annihilation rules are indexed with respect to agents, a detour in a proof consisting of K-import immediately followed by K-export (or vice versa) can only be eliminated if both rules pertain to the same agent:

$$\hat{k}_a\check{k}_a M \;\Rightarrow\; M \qquad \check{k}_a\hat{k}_a M \;\Rightarrow\; M \qquad (\forall a \in \mathit{People})$$

That this is intuitively correct can be seen using the deduction example from the previous section.

1.	$G \vdash M : K_1K_2K_3\varphi$	
2.	$G \boxdot_1 \varepsilon \vdash \check{k}_1 M : K_2K_3\varphi$	(K-import 1)
3.	$G \boxdot_1 \varepsilon \boxdot_2 \varepsilon \vdash \check{k}_2(\check{k}_1 M) : K_3\varphi$	(K-import 2)
4.	$G \boxdot_1 \varepsilon \boxdot_2 \varepsilon \boxdot_3 \varepsilon \vdash \check{k}_3(\check{k}_2(\check{k}_1 M)) : \varphi$	(K-import 3)
5.	$G \boxdot_1 \varepsilon \boxdot_2 \varepsilon \vdash \hat{k}_3(\check{k}_3(\check{k}_2(\check{k}_1 M))) : K_3\varphi$	(K-export 4)
6.	$G \boxdot_1 \varepsilon \vdash \hat{k}_2(\hat{k}_3(\check{k}_3(\check{k}_2(\check{k}_1 M)))) : K_2K_3\varphi$	(K-export 5)
7.	$G \vdash \hat{t}_1(\hat{k}_2(\hat{k}_3(\check{k}_3(\check{k}_2(\check{k}_1 M))))) : K_2K_3\varphi$	(T-export 6)

This derivation is the type theoretical analogon to (one of the proofs in) the example in the previous section, first the proposition φ is stripped of all knowledge operators by applying K-import, yielding a proof (object) $\check{k}_3(\check{k}_2(\check{k}_1 M))$ for φ in a subordinate K_3-context. Then the desired knowledge operators are put back on in reverse order, by applying K-export whenever the operator with the index of the subordinate context is needed, and T-export when it is not.

However, this is not the most efficient proof of $K_2K_3\varphi$, and the $\hat{k}_a\check{k}_a$-reduction rule can be used to simplify it: $\hat{t}_1(\hat{k}_2(\hat{k}_3(\check{k}_3(\check{k}_2(\check{k}_1 M))))) \Rightarrow \hat{t}_1(\hat{k}_2(\check{k}_2(\check{k}_1 M))) \Rightarrow \hat{t}_1(\check{k}_1 M)$. The resulting proof object corresponds to a proof where instead of first stripping off K_1, K_2 and K_3 and then putting all but K_1 back on again, only one subordinate proof is used:

1.	$G \vdash M : K_1K_2K_3\varphi$	
2.	$G \boxdot_1 \varepsilon \vdash \check{k}_1 M : K_2K_3\varphi$	(K-import 1)
3.	$G \vdash \hat{t}_1(\check{k}_1 M) : K_2K_3\varphi$	(T-export 2)

1.	$K_1K_2K_3\varphi$	
	$\boxed{K_1}$	
2.	$K_2K_3\varphi$	(K-import 1)
3.	$K_2K_3\varphi$	(T-export 2)

After switching to an epistemic alternative of agent 1 (K-import), one already sees that $K_2K_3\varphi$. Hence using the veridicality of knowledge (T-export) suffices to obtain a proof of $K_2K_3\varphi$. As in the monological case, the $\hat{k}\check{k}$-rule minimalizes the modal depth of the proof: the K_2- and K_3-strict subordinate proofs that are not necessary to derive the desired conclusion are eliminated.

The above does not yet show that indexing the reduction rules is necessary: in a multi agent *mono-modal* system, where we have just the K-rules for the modal operator, the annihilations rules for $\check{k}$ and $\hat{k}$ are also sound when the indices are disregarded. However, if we strengthen this system with T and 4 and disregard the agent indices, annihilation is not sound.

1.	$G \vdash M : K_1 K_2 \varphi$		$\check{4}_2(\hat{\imath}_1(\check{k}_1 M)) \Rightarrow \check{k}_1 M$
2.	$G \boxdot_1 \varepsilon \vdash \check{k}_1 M : K_2 \varphi$	(K-import 1)	($\hat{\imath}\check{4}$-reduction)
3.	$G \vdash \hat{\imath}_1(\check{k}_1 M) : K_2 \varphi$	(T-export 2)	
4.	$G \boxdot_2 \varepsilon \vdash \check{4}_2(\hat{\imath}_1(\check{k}_1 M)) : K_2 \varphi$	(4-import 3)	$G \boxdot_2 \varepsilon \vdash (\check{k}_1 M) : K_2 \varphi$

If the agent-index is disregarded the proofterm $\check{4}_2(\hat{\imath}_1(\check{k}_1 M))$ for $K_2\varphi$ $\check{4}\hat{\imath}$-reduces to $\check{k}_1 M$, and should therefore prove $K_2\varphi$ in context $G \boxdot_2 \varepsilon$. However, that this cannot be the case follows from the different agent indices of the separator and the proof function: they indicate that a K_1-formula was K-imported into a K_2-subordinate proof:

1.	$K_1 K_2 \varphi$	
2.	$[K_1]$ $K_2\varphi$	(K-import 1)
3.	$K_2\varphi$	(T-export 2)
4.	$[K_2]$ $K_2\varphi$	(4-import 3)

1.	$K_1 K_2 \varphi$	
2.	$[K_2]$ $K_2\varphi$	(K-import 1 ?)

Hence a term representing (part of) a correct natural deduction proof is turned into a term representing an incorrect proof.

Similar considerations make it evident that the modal reduction rules should also respect the operator-index. For the annihilation reductions formulated with respect to the indices, the results found earlier carry over to $MPTS_\beta^{People, Modalities}$.

4.2.5. THEOREM. *For all λ_S in the Modal Logic Cube of $MPTS_\beta^{People, Modalities}$ with added reduction rules:*

$$\check{k}_a^o \hat{k}_a^o M \Rightarrow M \textit{ for all } a \in People,\ o \in Modalities$$

$$\hat{k}_a^o \check{k}_a^o M \Rightarrow M \textit{ for all } a \in People,\ o \in Modalities$$

$$\check{4}_a^o \hat{\imath}_a^o M \Rightarrow M \textit{ for all } a \in People,\ o \in Modalities$$

$$\hat{\imath}_a^o \check{4}_a^o M \Rightarrow M \textit{ for all } a \in People,\ o \in Modalities$$

$$\check{5}_a^o \hat{\imath}_a^o M \Rightarrow M \textit{ for all } a \in People,\ o \in Modalities$$

$$\hat{\imath}_a^o \check{5}_a^o M \Rightarrow M \textit{ for all } a \in People,\ o \in Modalities$$

(and congruence rules) we have:

λ_S is Strongly Normalizing

λ_S has the Church Rosser property

λ_S enjoys the Subject Reduction property

PROOF. By repeating the proofs from chapter 3, the indexing does not interfere with the original rules.

The multi-agent multi-modal system also offers new possibilities for making detours in proofs and eliminating them. A first indication of this is that the two forced import rules describing the interaction in CD_{KB} are not completly independent. In a system where $F4$ holds for the pair of operators (o, o') and T-export for o, the FK-rule for (o, o') is derivable:

$$
\begin{array}{lll}
G \vdash M : \Box_a^o \varphi & G \vdash M : \Box_a^o \varphi & \\
G \boxdot_a^o \varepsilon \vdash \check{j}_a^o M : \varphi & G \boxdot_a^{o'} \varepsilon \vdash \check{g}_a^{o,o'} M : \Box_a^o \varphi & (F4\text{-import}) \\
 & G \boxdot_a^{o'} \varepsilon \boxdot_a^o \varepsilon \vdash \check{k}_a^o(\check{g}_a^{o,o'} M) : \varphi & (K\text{-import}) \\
 & G \boxdot_a^{o'} \varepsilon \vdash \hat{t}_a^o(\check{k}_a^o(\check{g}_a^{o,o'} M)) : \varphi & (T\text{-export}) \\
FK\text{-import} & \text{as a derived rule} &
\end{array}
$$

In these circumstances a new detour in proofs becomes possible:

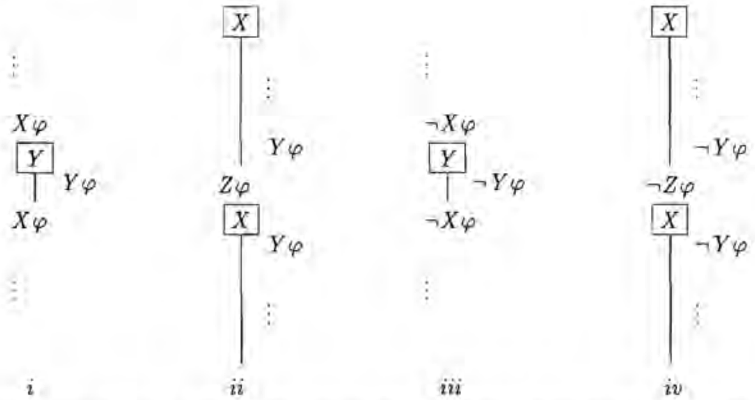
$$
\begin{array}{ll}
G \vdash M : \Box_a^o \varphi & \\
G \boxdot_a^{o'} \varepsilon \vdash \check{g}_a^{(o,o')} M : \Box_a^o \varphi & (F4\text{-import}) \\
G \vdash \hat{t}_a^{o'}(\check{g}_a^{(o,o')} M) : \Box_a^o \varphi & (T\text{-export})
\end{array}
$$

To eliminate this detour, a new annihilation rule could be added:

$$
\hat{t}_a^o(\check{g}_a^{o,o'} M) \Rightarrow M \qquad \forall a \in \mathit{People},\ \forall (o, o') \in O_{F4} \subseteq (\mathit{Modalities} \times \mathit{Modalities})
$$

In KB_{CD} this rule would remove combinations of FK-import of a B_a-formula into a K_a-strict subordinate proofs immediately followed by T-export. In terms of inspection formulas we can think of this detour as caused by the presence of both positive introspection ($B_a\varphi \supset K_a B_a \varphi$) and positive extraspection ($K_a B_a \varphi \supset B_a \varphi$) for K_a and B_a.

This observation can be extended to the format of general inspection principles. Whenever a system has pairs of positive introspection and extraspection principles $XY\varphi \supset Z\varphi$ and $YZ\varphi \supset X\varphi$, or negative introspection and extraspection principles $X\neg Y\varphi \supset \neg Z\varphi$ and $Y\neg Z\varphi \supset \neg X\varphi$, new detours become possible:



Although these detours can involve up to three different modal operators, they can be eliminated using rules of a familiar format. Assuming for the moment that X, Y and Z have the same agent-index, we define the following annihilations for pairs of positive introspection/extraspection and negative introspection/extraspection rules:

i $\hat{p}^{(X,Y,Z)}(\check{p}^{(Y,Z,X)}M) \Rightarrow M \qquad \forall X, Y, Z \in \mathit{Modalities}$

ii $\check{p}^{(X,Y,Z)}(\hat{p}^{(Y,Z,X)}M) \Rightarrow M \qquad \forall X, Y, Z \in \mathit{Modalities}$

iii $\hat{n}^{(X,Y,Z)}(\check{n}^{(Y,Z,X)}M) \Rightarrow M \qquad \forall X, Y, Z \in \mathit{Modalities}$

iv $\check{n}^{(X,Y,Z)}(\hat{n}^{(Y,Z,X)}M) \Rightarrow M \qquad \forall X, Y, Z \in \mathit{Modalities}$

Chapter 5

Modal predicate logic

Besides the strengthening of the modal rules, the expressive power of modal logic can also be strengthened by building it on a richer underlying language: predicate logic. This allows us to talk about individuals, their properties and relations.

However, modal predicate logic (MPL) is not simply 'modal logic with added quantifiers', there are a great number of choices to be made with respect to the behaviour of the quantifiers, identity, predicates and terms across possible worlds. All these choices yield different systems of MPL which have been charted and studied in model theoretical semantics (see for instance [Garson 1984]). The state of the art in natural deduction for MPL is definitely less impressive, for many systems of MPL no deductive counterparts are known.

In this chapter we first show how the construction for interpreting modal propositional logic can be extended to modal predicate logic. Then we concentrate on some of the well-known problems regarding the interaction between the modal operators, quantifiers and identity in the setting of MPTSs.

5.1. The interpretation

In this section we will indicate how the interpretation of modal propositional logic given earlier can be repeated for modal predicate logic. As before we will start by defining a typed system (ΛL) as close as possible to the original logic (L) and then showing that this system is equivalent to the system (λL) in the 'Modal Logic Cube'.

To accommodate the underlying *classical* first order predicate logic we will start the mapping by reformulating our modal logic as a second order intuitionistic predicate logic $\Box PRED2$ and then indicate how the path:

$$\Box PRED2 \Leftrightarrow \Lambda\Box PRED2 \Leftrightarrow \lambda\Box PRED2$$

to the MPTS $\lambda\Box PRED2$ through the intermediate system $\Lambda\Box PROP2$ could be followed.

An additional difficulty compared to propositional logic is that for predicate logics the typed lambda calculus style of presentation is rather different from the usual logical presentation: functions and predicates are formed by λ-abstraction. However, $PRED2$ is conservative over the system $PRED^{-fr}$ which has constants for functions and predicates as first order predicate logic in the regular formulation. Hence, the complete picture of the interpretation looks like:

$$\Box PRED2^{-fr} \Rightarrow \Box PRED2 \Leftrightarrow \Lambda\Box PRED2 \Leftrightarrow \lambda\Box PRED2$$

where the $\Rightarrow$ is the conservativity of $\Box PRED2$ over $\Box PRED2^{-fr}$.

5.1.1. $\Box PRED2$

$\Box PRED2$ is based on the second order intuitionistic predicate logic $PRED2$ from the Logic Cube as defined in [Geuvers 1993].

1 The *domains* are given by

$$\mathcal{D} ::= \mathcal{B} | Prop | (\mathcal{D} \rightarrow \mathcal{D}),$$

where $\mathcal{B}$ is a specific set of *basic domains*.

Hence $\Box PRED2$ is a many sorted logic from the start.

2 The *order of the domain*, $\text{ord}(D)$, is defined by

$$\begin{aligned}
\text{ord}(B) &= 1 \; \textit{for} \; B \in \mathcal{B} \\
\text{ord}(Prop) &= 2 \\
\text{ord}(D_1 \rightarrow \ldots \rightarrow D_p \rightarrow B) &= max\{\text{ord}(D_i) | 1 \leq i \leq p\}, \; \textit{if} \; B \in \mathcal{B} \\
\text{ord}(D_1 \rightarrow \ldots \rightarrow D_p \rightarrow Prop) &= max\{\text{ord}(D_i) | 1 \leq i \leq p\} + 1
\end{aligned}$$

For $\Box PRED2$, $\text{ord}(D) \leq 2$, hence $max\{\text{ord}(D_i)|1 \leq i \leq p\} \leq$ *if* $B \in \mathcal{B}$ and $max\{\text{ord}(D_i)|1 \leq i \leq p\} + 1 \leq 2$. The second condition shows that domains 'ending' in $Prop$ can not have D_i containing (or being) $Prop$.

3 The *terms* of the second order language are described as follows:

- There are countably many variables of domain D if $\text{ord}(D) \leq 2$,
- If $M \in D_2$, x a variable of domain D_1 and $\text{ord}(D_1 \rightarrow D_2) \leq 2$, then $\lambda x \in D_1.M \in D_1 \rightarrow D_2$,
- If $M \in D_1 \rightarrow D_2$, $N \in D_1$, then $MN \in D_2$,
- If $\varphi \in Prop$, x a variable of domain D with $\text{ord}(D) \leq 2$, then $\forall x \in D.\varphi \in Prop$,
- If $\varphi \in Prop$ and $\psi \in Prop$, then $\varphi \supset \psi \in Prop$
- If $\varphi \in Prop$, then $\Box\varphi \in Prop$

Using the second clause, predicates can be defined by λ-abstraction: given a set (domain) A and a formula φ, $\lambda x \in A.\varphi$ is a term in $(A \rightarrow PROP)$, the predicates over A.

4 On the terms we have definitional equality by β-conversion. This equality is denoted by $=_\beta$. Terms φ for which $\varphi \in Prop$ are *formulas*, 'Form' denotes the set of formulas.

The underlying predicate logic of $\Box PRED2$ is *second order intuitionistic* logic. This requires a few simple modification of the deduction rules given for $\Box PROP2$: the $\forall$-rules and hold for the variables of all domains D not just $Prop$, likewise the term rule deals with all terms $t \in D$. All other rules are the same.

5 For φ and ψ formulas of the second order language the Fitch-style *deduction rules* of $\Box PRED2$ are:

⊃-intro

$$\begin{array}{l} \left|\begin{array}{l} \varphi \\ \hline \vdots \\ \psi \end{array}\right. \\ \varphi \supset \psi \end{array}$$

⊃-elim

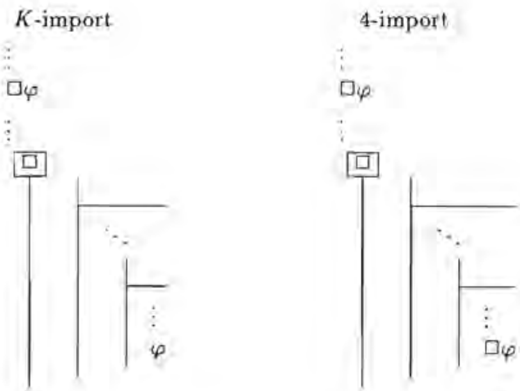
$$\begin{array}{l} \vdots \\ \varphi \supset \psi \\ \vdots \\ \varphi \\ \vdots \\ \psi \end{array}$$

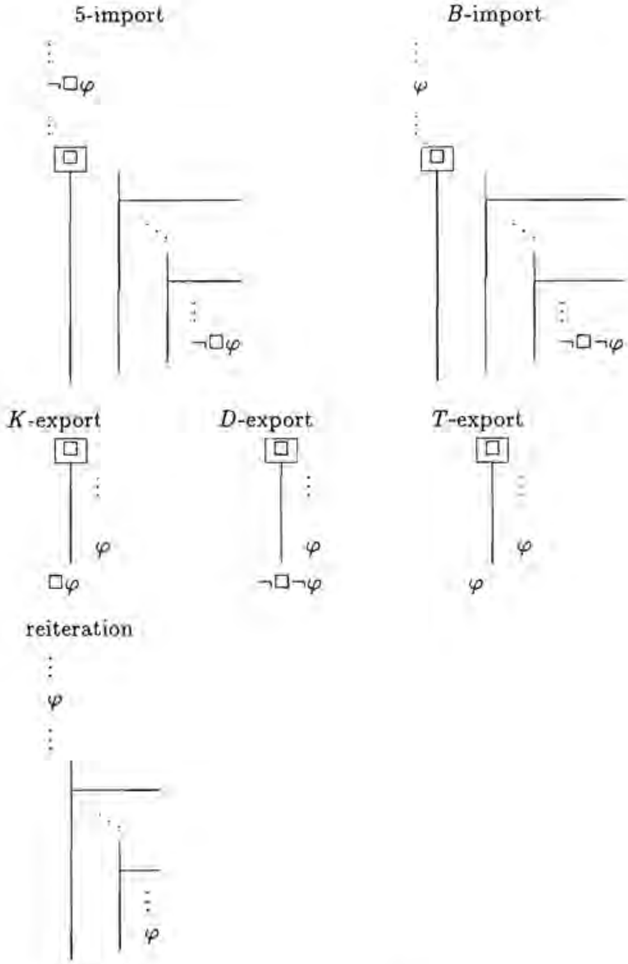
∀-intro

$$\begin{array}{l} \vdots \\ \left|\begin{array}{l} x \in D \\ \hline \vdots \\ \varphi \end{array}\right. \\ \forall x \in D.\varphi \end{array}$$

∀-elim

$$\begin{array}{l} \vdots \\ \forall x \in D.\varphi \\ \vdots \\ t \in D \\ \vdots \\ \varphi[t/x] \end{array}$$





term	axiom	conversion
$t \in D$	$\forall x \in Prop.((x \supset \bot) \supset \bot) \supset x$	$\vdots$ ψ $\varphi \quad (\varphi =_\beta \psi)$

The only rule that is new compared to the deduction rules for $\square PROP2$ is the conversion rule (*conv*), which is applied in the following way.

5.1.1. DEFINITION. **Application of deduction rules**

$R = conv$.
The premiss P and the conclusion E have to lie in the same interval and $P =_\beta E$.

The role of conversion in the logic $PRED2$ is twofold:

Function application: in $PRED2$ we can construct functions (and predicates) by λ-abstraction, e.g. $(\lambda x \in D_1.t) \in D_1 \to D_2$. In working with these functions, one wants to be able to exchange a function-argument pair occuring in a formula $(\varphi((\lambda x \in D.t)q))$ with the calculated result $(\varphi(t[q/x]))$. This can be done by means of conversion, since the pair and the result are β-equal $(\varphi((\lambda x \in D.t)q) =_\beta \varphi(t[q/x]))$.

Comprehension: the property that given any formula φ (with free variables $\vec{x}$) that describes a certain part of the domain, there exists an n-ary predicate $\chi\vec{x}$ that does the same. This property is often expressed in the form of an axiom:

$$\forall\varphi\exists\chi.(\forall\vec{x}.\varphi \leftrightarrow \chi\vec{x}).$$

Comprehension is derivable in $PRED2$. Given a formula, say ψ with free variables $x \in D$ and $y : D'$, we can easily construct the predicate by means of λ-abstraction: $\lambda x \in D\lambda y \in D'.\psi$. Conversion ensures that the equivalence between φ and $\chi\vec{x}$ stated in the axiom holds $(\psi =_\beta (\lambda x \in D\lambda y \in D'.\psi)xy)$.

Connectives

Like $\Box PROP2$, $\Box PRED2$ is a second order system. Hence we again define all the usual connectives in terms of $\forall$ and $\supset$, the existential quantifier is defined as follows (let φ be a formula):

$$\exists x^D.\varphi := \forall\alpha \in Prop.(\forall x \in D.\varphi \supset \alpha) \supset \alpha.$$

As before we add the double negation axiom to obtain classical predicate logic.

Extensionality

In $\Box PRED2$, terms of domain $D \to Prop$ are to be understood as predicates on D or as subsets of D. Hence from a set theoretical point of view we would want to identify predicates that are extensionally equal:

$$(\forall\vec{x}.f\vec{x} \supset g\vec{x} \wedge \forall\vec{x}.g\vec{x} \supset f\vec{x}) \supset f =_D g$$

In general it is not possible to express extensionality in the logic, hence Geuvers adds a *schematic rule of extensionality*:

$$(EXT)\ \frac{f\vec{x} \supset g\vec{x} \quad g\vec{x} \supset f\vec{x} \quad \varphi(f)}{\varphi(g)}(*)$$

where (for $\Box PRED2$) f and g are arbitrary terms of the same domain D ($(*)$ signifies the restriction that variables of x may not occur free in a non-discharged assumption of the derivations in the premisses of the rule). E-PRED2 is the system PRED2 extended with (EXT).

It is an open question whether adding the schema (EXT) to $(\Box)PRED2$ to obtain E-$(\Box)PRED2$ actually strengthens $(\Box)PRED2$. Obviously E-$(\Box)PRED2$ is conservative over $(\Box)PRED2$, simply do not use the EXT-rule. The question is whether there is conservativity in the other direction: $E\text{-}(\Box)PRED2 \Rightarrow (\Box)PRED2$. The conjecture is that conservativity holds. Counterexamples usually involve pairs of formulas like $Q(\lambda\alpha : Prop.\alpha \supset \alpha)$ and $Q(\lambda\alpha : Prop.\alpha \supset \neg\neg\alpha)$, where EXT is used to derive one from the other, whereas this is impossible in the non-extensional system. However, in this counterexample the order of the formulas is ≥ 3 and hence it can not be used as a counterexample for the conservativity of $E\text{-}PRED2$ over $PRED2$. Probably this argument can be generalized to any application of EXT in $E\text{-}PRED2$:

$$\frac{\varphi(P_1) \quad P_1\vec{x} \supset P_2\vec{x} \quad P_2\vec{x} \supset P_1\vec{x}}{\varphi(P_2)}$$

In order for $\varphi(P_1)$ (or $(\varphi(P_2))$) to be of order ≤ 2, P_1 has to be in 'fully applicative form', which means that all argument places in P_1 have to be filled (it is quantor free). In that case $\varphi(P_2)$ can already be derived form $\varphi(P_1)$ in $PRED2$. If this argument is correct all counterexamples fail and hence we have $E\text{-}PRED2 \Leftrightarrow PRED2$.

$\Box PRED2^{-fr}$

The definition of the underlying predicate logic of $\Box PRED2$ is rather different from the traditional presentation of (first order) predicate logic: predicates as well as functions are defined using λ-abstraction. In [Geuvers 1993] a system $PRED^{-fr}$ is defined with constants for predicates and functions is given, which is proved to be conservative over the first order minimal predicate logic $PRED$.

Adopting this idea, we could start the interpretation of modal predicate logic from a system $\Box PRED2^{-fr}$ which is much closer to the conventional presentation modal predicate logic, and which should be conservative over $\Box PRED2$. This system is introduced in two stages, first the possibility of defining functions by abstraction is removed then that of creating 'abstraction predicates'.

$\Box PRED2$ is conservative over $\Box PRED2^{-f}$, a version of $\Box PRED2$ which has only the simplest ('first order') domains for functions. The language of $\Box PRED2^{-f}$ is defined as follows.

1 The *functional domains* are given by

$$F ::= \mathcal{B} \to \ldots \to \mathcal{B},$$

where $\mathcal{B}$ is a specific set of *basic domains*. (Every functional domain has to be built up from at least two basic domains)

2 The *domains* are given by

$$\mathcal{D} ::= \mathcal{B}|Prop|\mathcal{D} \to \ldots \to \mathcal{D} \to Prop,$$

3 The *order of the domain*, $\text{ord}(D)$, is defined by

$$\begin{aligned} \text{ord}(B) &= 1 \text{ for } B \in \mathcal{B} \\ \text{ord}(Prop) &= 2 \\ \text{ord}(D_1 \to \ldots \to D_p \to B) &= max\{\text{ord}(D_i)|1 \leq i \leq p\}, \text{ if } B \in \mathcal{B} \\ \text{ord}(D_1 \to \ldots \to D_p \to Prop) &= max\{\text{ord}(D_i)|1 \leq i \leq p\} + 1 \end{aligned}$$

Note that functional domains all have order 1.

4 There are countably many function-constants c_i^F for every function domain $F \in \mathcal{F}$ in $\Box PRED2^{-f}$.

5 The *terms* of the second order language are described as follows:

- There are countably many variables of domain D if $\text{ord}(D) \leq 2$,
- If c_i^F is a function constant of domain $F \equiv B_1 \to \ldots \to B_{p+1}$ and $t_i \in B_i$ for $1 \leq i \leq p$, then $c_i^F t_1, \ldots, t_p \in B_{p+1}$,
- If $t \in D_2$, x a variable of domain D_1 and $\text{ord}(D_1 \to D_2) \leq 2$, then $(\lambda x \in D_1.t) \in D_1 \to D_2$,
- If $t \in D_1 \to D_2$, $q \in D_1$, then $tq \in D_2$,
- If $\varphi \in Prop$, x a variable of domain D with $\text{ord}(D) \leq 2$, then $(\forall x \in D.\varphi) \in Prop$,
- If $\varphi \in Prop$ and $\psi \in Prop$, then $\varphi \supset \psi \in Prop$
- If $\varphi \in Prop$,then $\Box\varphi \in Prop$

6 The deduction rules are the same as those of $\Box PRED2$, the quantification is restricted to (basic) domains since we have no variables over function domains.

The next step is to eliminate the possibility to form predicates by means of λ-abstraction: $\Box PRED2^{-fr}$ is $\Box PRED2^{-f}$ where the definition of terms is changed to

- There are countably many variables of domain D if $\text{ord}(D) \leq 2$,
- If c_i^F is a function constant of domain $F \equiv B_1 \to \ldots \to B_{p+1}$ and $t_i \in B_i$ for $1 \leq i \leq p$, then $c_i^F t_1, \ldots, t_p \in B_{p+1}$,
- If $t \in D_1 \to \ldots \to D_p \to Prop$, $q_i \in D_i$ for $1 \leq i \leq p$, then $tq_1 \ldots q_p \in Prop$,
- If $\varphi \in Prop$, x a variable of domain D with $\text{ord}(D) \leq 2$, then $\forall x \in D.\varphi \in Prop$,
- If $\varphi \in Prop$ and $\psi \in Prop$, then $\varphi \supset \psi \in Prop$.

This definition of the language corresponds to the way in which first order predicate logic is usually set up: the terms of the object language are inductively defined from variables and constants by function application, and the set of formulas is inductively defined from the basic formulas by applying connectives (basic formulas are of the form $x^D t_1 \ldots t_p$, with t_i terms of the object language, and allowing for $p = 0$).

Note that in this logic conversion can no longer play a role, we cannot build functions or predicates by abstraction and hence it is no longer possible to calculate the result of applying such a function or predicate to an argument by β-conversion. For the same reason we cannot construct the comprehension relation for an arbitrary universal formula, the needed abstraction is no longer allowed. This is what one would expect from a logic close to the traditional presentation of first order predicate logic, where conversion is not an inherent phenomenon.

5.1.2. $\Lambda\Box PRED2$

The typed λ-calculus $\Lambda\Box PRED2$ corresponding to the logic $\Box PRED2$ is defined in the same way as $\Lambda\Box PROP2$ in chapter 3. The main difference is that the language of $\Lambda\Box PRED2$ is richer than that of $\Lambda\Box PROP2$, for which the set of *functional types* was empty and the set of *predicate types* of consisted solely of *Prop*.

1 The set of *functional types* of $\Lambda\Box PRED$, $Type^f$, is defined by the following abstract syntax.

$$Type^f ::= Var^{ty} | Type^f \rightarrow Type^f,$$

where Var^{ty} is a countable set of *type-variables*.

The set of *predicate types* of $\Lambda\Box PRED2$, $Type^p$, consists of the expressions

$$\sigma_1 \rightarrow \sigma_2 \rightarrow \ldots \rightarrow Prop,$$

with $n \geq 0$ and all σ_i functional types.

2 The *object-terms* of the language of $\Lambda\Box PRED2$ form a subset of the set of *pseudoterms*, T, which is generated by the following abstract syntax.

$$T ::= Var^{ob} | TT | \lambda x : Type^f.T | T \supset T | \forall Var^{ob} : Type^f.T | \forall Var^{ob} : Type^p.T | \Box T,$$

with Var^{ob} a countable set of *object-variables*. An object-term is of a certain type only under assumption of specific types (functional or predicative) for the free variables that occur in that term. That the object term t *is of type* A if x_i is of type A_i for $1 \leq i \leq n$, is denoted by the judgement

$$x_1 : A_1, \ldots, x_i : A_i; \Delta_1 \boxdot \ldots \boxdot x_j : A_j, \ldots, x_n : A_n; \vdash t : A.$$

Here $x_1, \ldots, x_n$ are different object variables and $A_1, \ldots A_n$ are types. As before we call such a sequence of statements and separators a *generalized object context*, and we let $F, F', \ldots$ range over the set of generalized object contexts. These contexts are also represented as $\Gamma_1 \boxdot \ldots \boxdot \Gamma_n$, where $\Gamma_1, \ldots, \Gamma_n$ range over the set of *object contexts* (sequences of declarations of the form $x_i : A_i$, uninterrupted by separators). The rules for deriving judgements are the following.

$$(var)\ \frac{}{\Gamma_1 \boxdot \ldots \boxdot \Gamma_n \vdash x : A} \quad \textit{if } x : A \textit{ in } \Gamma_n$$

$$(\lambda\, abs)\ \frac{F, x : A \vdash t : B}{F \vdash \lambda x : A.t : A \rightarrow B} \quad \textit{if } A \textit{ a functional or predicate type}$$

$$(app)\ \frac{F \vdash q : A \rightarrow B \quad F \vdash t : A}{F \vdash qt : B}$$

$$(\supset)\ \frac{F \vdash \varphi : Prop \quad F \vdash \psi : Prop}{F \vdash \varphi \supset \psi : Prop}$$

$$(\forall)\ \frac{F, x : A \vdash \varphi : Prop}{F \vdash (\forall x : A.\varphi) : Prop} \quad \textit{if } A \textit{ a functional or predicate type}$$

$$(\Box)\ \frac{F \vdash \varphi : Prop}{F \vdash \Box\varphi : Prop}$$

$$(transfer)\ \frac{F \vdash \varphi : A}{F \boxdot \varepsilon \vdash \varphi : A} \quad \textit{if } A \textit{ a functional or predicate type}$$

Note that compared to $\Lambda\Box PROP2$ we have additional rules (λabs, (app)) for the formation of functions and predicates by means of abstraction and application. As before the (*transfer*)-rule allows the use of any non-proof term derivable on a context G in arbitrary (deep) subordinate contexts of G.

3 The set of *proof-terms* is a subset of the set of *pseudoproofs*, Pr, generated by the following abstract syntax.

$$Pr ::= Var^{Pr} | PrPr | PrT | \lambda x : Type.Pr | \lambda x : T.Pr | \check{k}Pr | \bar{4}Pr | \breve{5}Pr | \bar{b}Pr | \bar{k}Pr \; | \bar{d}Pr \; | \hat{t}Pr \; | C,$$

where Var^{Pr} is the set of proof-variables (C, the set of constants).

That the proof term M is of type A if p_i is of type φ_i for $1 \leq i \leq l$, is denoted by

$$\Gamma_1; p_1 : \varphi_1, \ldots, p_h : \varphi_h \boxdot \ldots \boxdot \Gamma_n; p_k : \varphi_k, \ldots, p_l : \varphi_l \vdash M : A$$

where the $\Gamma_1, \ldots, \Gamma_n$ are as in 2, $p_1, \ldots p_l$ are different proof-variables and

$$\Gamma_1 \boxdot \ldots \boxdot \Gamma_n \vdash \varphi_j : Prop \quad for\ k \leq j \leq l,$$

We call such a sequence of statements and separators a *generalized context* and let $G, G', \ldots$ range over the set of generalized contexts. A *proof-context* is an uninterrupted sequence of declarations with different proof variables as subject ($p_1 : \varphi_1, \ldots, p_h : \varphi_h$). By letting $\Delta_1, \Delta_2, \ldots$ range over the set of proof contexts, generalized contexts can be represented as being of the form $\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n$.

The rules for deriving judgements are the following.

$$(axiom)\ \frac{}{\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n \vdash p : \varphi} \quad if\ p : \varphi\ in\ \Delta_n$$

$$(\supset intro)\ \frac{G, p : \varphi \vdash M : \psi}{G \vdash (\lambda p : \varphi.M) : \varphi \supset \psi}$$

$$(\supset elim)\ \frac{G \vdash M : \varphi \supset \psi \quad G \vdash N : \varphi}{G \vdash MN : \psi}$$

$$(\forall\, intro)\ \frac{\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n, x : A; \Delta_n \vdash M : \varphi}{\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n \vdash (\lambda x : A.M) : \forall x : A.\varphi} \quad if\ x \notin FV(\Delta_n)$$

$$(\forall\, elim)\ \frac{\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n \vdash M : (\forall x : A.\varphi) \quad \Gamma_1 \boxdot \ldots \boxdot \Gamma_n \vdash t : A}{\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n \vdash Mt : \varphi[t/x]}$$

$$(conv)\ \frac{\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n \vdash M : \varphi \quad \Gamma_1 \boxdot \ldots \boxdot \Gamma_n \vdash \psi : Prop}{\Gamma_1; \Delta_1 \boxdot \ldots \boxdot \Gamma_n; \Delta_n \vdash M : \psi} \quad if\ \varphi =_\beta \psi$$

$$(K\ import)\ \frac{G \vdash M : \Box\varphi\ (: Prop)}{G \boxdot \varepsilon; \varepsilon \vdash \check{k}M : \varphi}$$

$$(4\ import)\ \frac{G \vdash M : \Box\varphi\ (: Prop)}{G \boxdot \varepsilon; \varepsilon \vdash \bar{4}M : \Box\varphi}$$

$$(5\ import)\ \frac{G \vdash M : \neg\Box\varphi\ (: Prop)}{G \boxdot \varepsilon; \varepsilon \vdash \breve{5}M : \neg\Box\varphi}$$

$$(B\ import)\ \frac{G\ \vdash\ M : \varphi\ (: Prop)}{G \boxdot \varepsilon;\varepsilon\ \vdash\ \check{b}M : \neg\Box\neg\varphi}$$

$$(K\ export)\ \frac{G \boxdot \varepsilon;\varepsilon\ \vdash\ M : \varphi\ (: Prop)}{G\ \vdash\ \hat{k}M : \Box\varphi}$$

$$(D\ export)\ \frac{G \boxdot \varepsilon;\varepsilon\ \vdash\ M : \varphi\ (: Prop)}{G\ \vdash\ \hat{d}M : \neg\Box\neg\varphi}$$

$$(T\ export)\ \frac{G \boxdot \varepsilon;\varepsilon\ \vdash\ M : \varphi\ (: Prop)}{G\ \vdash\ \hat{t}M : \varphi}$$

$$(Doubleneg)\ \varepsilon;\varepsilon\ \vdash\ c : (\forall\alpha : Prop.((\alpha \supset \bot) \supset \bot) \supset \alpha)$$

$$(Transfer_{\neg\neg})\ \frac{G\ \vdash\ c : (\forall\alpha : Prop.((\alpha \supset \bot) \supset \bot) \supset \alpha)}{G \boxdot \varepsilon;\varepsilon\ \vdash\ c : (\forall\alpha : Prop.((\alpha \supset \bot) \supset \bot) \supset \alpha)}$$

With respect to the meta-theoretical properties of $\Lambda\Box PRED2$ that are needed in proving the soundness of the interpretation, the only difference with $\Lambda\Box PROP2$ is that we have to take conversion into account when stating Stripping (like in the Stripping Lemma for the Modal Logic Cube): let $G\ \vdash\ M : \varphi$, where $G \equiv \Gamma_1;\Delta_1 \boxdot \Gamma_n;\Delta_n$ (and $F \equiv \Gamma_1 \boxdot \ldots \boxdot \Gamma_n$) be derivable in $\Lambda\Box PRED2$.

$\Gamma_1 \boxdot \ldots \boxdot \Gamma_n\ \vdash\ x : A$ (x *an object variable*) $\Rightarrow$
$A = B$ *with* $x : B \in \Gamma_i$ *for some* i, $1 \leq i \leq n$

$F\ \vdash\ \chi : Prop$ *and* $\chi = \varphi \supset \psi\ \Rightarrow$
$F\ \vdash\ \varphi : Prop$ *and* $F\ \vdash\ \psi : Prop$

$\Gamma_1 \boxdot \ldots \boxdot \Gamma_n\ \vdash\ \forall x : Prop.\psi : Prop\ \Rightarrow$
$\Gamma_1 \boxdot \ldots \boxdot \Gamma_n, x : Prop\ \vdash\ \psi : Prop$

$F\ \vdash\ \Box\varphi : Prop\ \Rightarrow\ F\ \vdash\ \varphi : Prop$

$\Gamma_1;\Delta_1 \boxdot \ldots \boxdot \Gamma_n;\Delta_n\ \vdash\ p : \varphi$ (p *a proof variable*) $\Rightarrow\ \psi = \varphi$ *with* $p : \varphi \in \Delta_n$

$\Gamma_1;\Delta_1 \boxdot \ldots \boxdot \Gamma_n;\Delta_n\ \vdash\ \lambda x : A.M : \varphi\ \Rightarrow$
$\Gamma_1;\Delta_1 \boxdot \ldots \boxdot \Gamma_n, x : A;\Delta_n\ \vdash\ M : \psi$
with $\varphi = \forall x : A.\psi$ *for some* ψ

$\Gamma_1;\Delta_1 \boxdot \ldots \boxdot \Gamma_n;\Delta_n\ \vdash\ \lambda p : \chi.M : \varphi$ (χ *a proposition*) $\Rightarrow$
$\Gamma_1;\Delta_1 \boxdot \ldots \boxdot \Gamma_n, p : \chi;\Delta_n\ \vdash\ M : \psi$ *with* $\varphi = \chi \supset \psi$ *for some* ψ

$G\ \vdash\ MN : \varphi$ (N *a proof*) $\Rightarrow\ G\ \vdash\ M : \psi \supset \chi$ *and* $G\ \vdash\ N : \psi$
with $\varphi = \chi$ *for some* ψ, χ

$\Gamma_1;\Delta_1 \boxdot \ldots \boxdot \Gamma_n;\Delta_n\ \vdash\ Mt : \varphi$ (t *an object*) $\Rightarrow$
$\Gamma_1;\Delta_1 \boxdot \ldots \boxdot \Gamma_n;\Delta_n\ \vdash\ M : \forall x : A.\psi$, *and* $\Gamma_1 \boxdot \ldots \boxdot \Gamma_n\ \vdash\ t : A$
with $\varphi = \psi[t/x]$, *for some* ψ, φ

$G\ \vdash\ \check{k}M : \varphi$ (M *a proof*) $\Rightarrow\ G \equiv G' \boxdot \Gamma;\Delta$ *and* $G'\ \vdash\ M : \Box\psi$ *where* $\psi = \varphi$

$G\ \vdash\ \check{4}M : \varphi$ (M *a proof*) $\Rightarrow\ G \equiv G' \boxdot \Gamma;\Delta$ *and* $G'\ \vdash\ M : \Box\psi$ *where* $\Box\psi = \varphi$

$$G \vdash \check{5}M : \varphi\ (M\ a\ proof) \Rightarrow G \equiv G' \boxdot \Gamma; \Delta\ and\ G' \vdash M : \neg\Box\psi\ where\ \neg\Box\psi = \varphi$$

$$G \vdash \check{b}M : \varphi\ (M\ a\ proof) \Rightarrow G \equiv G' \boxdot \Gamma; \Delta\ and\ G' \vdash M : \psi\ where\ \neg\Box\neg\psi = \varphi$$

$$G \vdash \check{k}M : \varphi\ (M\ a\ proof) \Rightarrow G \boxdot \varepsilon; \varepsilon \vdash M : \psi\ where\ \Box\psi = \varphi$$

$$G \vdash \check{d}M : \varphi\ (M\ a\ proof) \Rightarrow G \boxdot \varepsilon; \varepsilon \vdash M : \psi\ where\ \neg\Box\neg\psi = \varphi$$

$$G \vdash \check{t}M : \varphi\ (M\ a\ proof) \Rightarrow G \boxdot \varepsilon; \varepsilon \vdash M : \psi\ where\ \psi = \varphi$$

$$G \vdash c : \varphi \Rightarrow \varphi = \forall\alpha : Prop.((\alpha \supset \bot) \supset \bot) \supset \alpha$$

5.1.3. Mapping $\Box PRED2$ to $\Lambda\Box PRED2$

The OK natural deduction proofs of $\Box PRED2$ (a proof Σ of φ in $\Box PRED2$ is an *OK proof* if all reiterated formulas in Σ are assumptions) are mapped to terms of $\Lambda\Box PRED2$ using a slightly modified version of the mapping '!' defined in chapter 2.

To an OK-deduction of $\varphi_1, \ldots, \varphi_n \vdash \psi$ in $\Box PRED2$ we are going to associate an object-context Γ and a proof-term M such that $\Gamma; p_1 : \varphi_1, \ldots, p_n : \varphi_n \vdash M : \psi$.

In order to make M a faithful representation of the deduction in $\Box PRED2$, Γ should assign types to all the free term-variables in the deduction that are not 'bound by a $\forall$' at any later stage.

5.1.2. DEFINITION. **Term-contexts**
For every term t of the language of $\Box PRED2$ we define a context Γ_t such that $\Gamma_t \vdash t : D$ (in $\Lambda\Box PRED2$) if $t \in D$ (in $\Box PRED2$), as follows

$$\begin{aligned}
t \equiv x^D &\Rightarrow \Gamma_t := x^D : D,\\
t \equiv \lambda x \in D.M &\Rightarrow \Gamma_t := \Gamma_M/(x : D),\\
t \equiv MN &\Rightarrow \Gamma_t := \Gamma_M \cup \Gamma_N,\\
t \equiv \varphi \supset \psi &\Rightarrow \Gamma_t := \Gamma_\varphi \cup \Gamma_\psi,\\
t \equiv \forall x \in D.\varphi &\Rightarrow \Gamma_t := \Gamma_\varphi/(x : D),\\
t \equiv \Box\varphi &\Rightarrow \Gamma_t := \Gamma_\varphi.
\end{aligned}$$

This definition is correct in the sense that every term t is derivable on 'its' context Γ_t.

5.1.3. DEFINITION. **'!', the mapping**
The OK-deductions in $\Box PRED2$ are mapped to a term context (Γ), object/proof-context (Δ) and a term (M) in $\Lambda\Box PRED2$ by the following inductively defined mapping '!' (assuming that in an OK-deduction all bound variables are chosen to be different in such a way that they differ from the free ones). '!' is the same as for $\Box PROP2$, except for the cases that involve object variables which can now be variables over an arbitrary domain D instead of *Prop* only:

var-assumption

$$x \in D^i$$

$$\Rightarrow$$

$$x_i : D\ \varepsilon \vdash x_i : D$$

∀-intro

$$\begin{array}{ll} \boxed{\begin{array}{l} x \in D' \\ \vdots \\ \psi \end{array}} & \Sigma \\ \forall x \in D.\psi & \end{array}$$

$$\Rightarrow$$

$$\frac{\Gamma_1;\Delta_1 \boxdot \ldots \boxdot \Gamma_n;\Delta_n \vdash \Sigma^! : \psi}{\Gamma_1 \boxdot \ldots \boxdot \Gamma_n/(x_i : Prop);\Delta_n \vdash \lambda x_i : D.\Sigma^! : \forall x : D.\psi}$$

∀-elim

$$\begin{array}{ll} \vdots & \\ t \in D & \\ & \Sigma \\ \vdots & \\ \forall x \in D.\psi & \\ \psi[t/x] & \end{array}$$

$$\Rightarrow$$

$$\frac{\Gamma_1;\Delta_1 \boxdot \ldots \boxdot \Gamma_n;\Delta_n \vdash \Sigma^! : \forall x : D.\psi \quad \Gamma_t \vdash t : D}{\Gamma_1 \cup^* \Gamma_t;\Delta_1 \boxdot \ldots \boxdot \Gamma_n;\Delta_n \vdash \Sigma^! t : \psi[t/x]}$$

And there is one new clause, since we now have conversion in the logic:

conversion

$$\begin{array}{ll} & \Sigma \\ \vdots & \\ \psi & (\varphi =_\beta \psi) \\ \varphi & \end{array}$$

$$\Rightarrow$$

$$\frac{G \vdash \Sigma^! : \psi}{G \vdash \Sigma^! : \varphi}$$

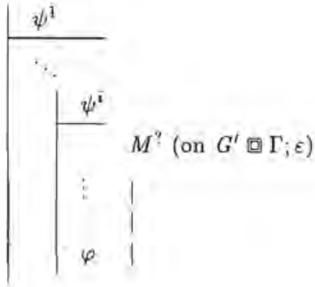
5.1.4. Mapping $\Lambda\Box PRED2$ to $\Box PRED2$

In the definition of the mapping '?' from lambda-terms back to natural deduction proofs we have to make some more changes, since in each case the last applied rule in the type derivation could have been conversion, which leaves the term unchanged but changes the type to a type that is β-equal. Therefore we let '?' map the terms of $\Lambda\Box PRED2$ to (partial) proof figures where the final rule is always an application of conversion, which can be vacuous.

As before, we start mapping a term after the statements in the rightmost proof context (Δ_n) have been turned into hypotheses of the pre-derivation.

5.1.4. DEFINITION. **Proof context mapping**

$G' \boxtimes \Gamma; p_1 : \psi_1, \ldots, p_i : \psi^i \vdash M : \varphi \Rightarrow$



5.1.5. DEFINITION. **'?', the mapping**
For any proof-terms M with $G \vdash M : \varphi$ we define by induction on the structure of M a pre-derivation $M^?$ of φ with respect to G as follows (where, for arbitrary n, $G \equiv \Gamma_1; \Delta_1 \boxtimes \ldots \boxtimes \Gamma_n; \Delta_n$, $G' \equiv \Gamma_1; \Delta_1 \boxtimes \ldots \boxtimes \Gamma_{n-1}; \Delta_{n-1}$, and $F \equiv \Gamma_1 \boxtimes \ldots \boxtimes \Gamma_n$).

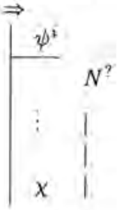
proof var
$G \vdash p_i : \varphi$
$\Rightarrow$
ψ^i
φ

object var
$F \vdash x_i : D$
$\Rightarrow$
$x \in D^i$

term
$F \vdash t : D$
$\Rightarrow$
$t \in D$

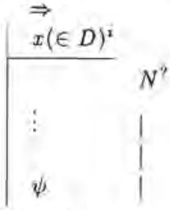
axiom constant
$G \vdash c : \forall x : D.((x \supset \bot) \supset \bot) \supset x$
$\Rightarrow$
$\forall x : Prop.((x \supset \bot) \supset \bot) \supset x$

proof-abstraction
$G \vdash \lambda p_i : \psi.N : \varphi$
$\Rightarrow$

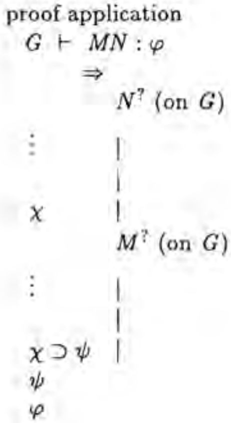


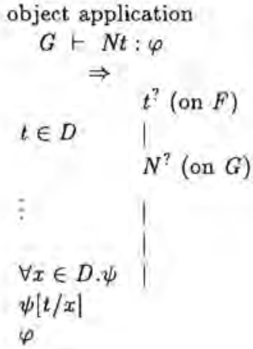
$\psi \supset \chi$
φ

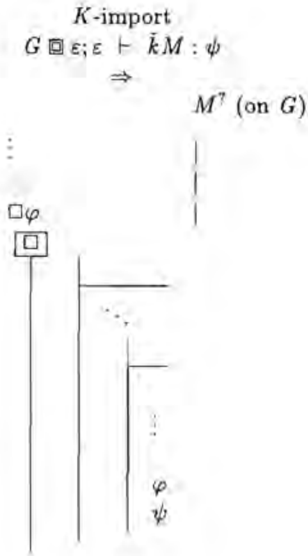
object-abstraction
$G \vdash \lambda x_i : D.N : \varphi$
$\Rightarrow$

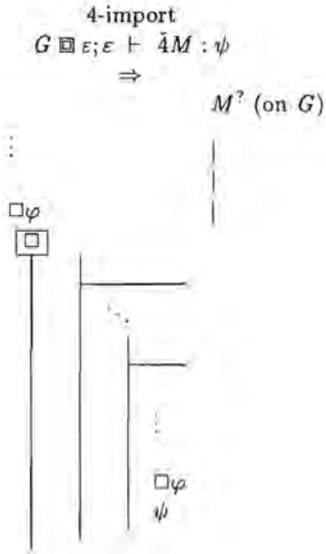


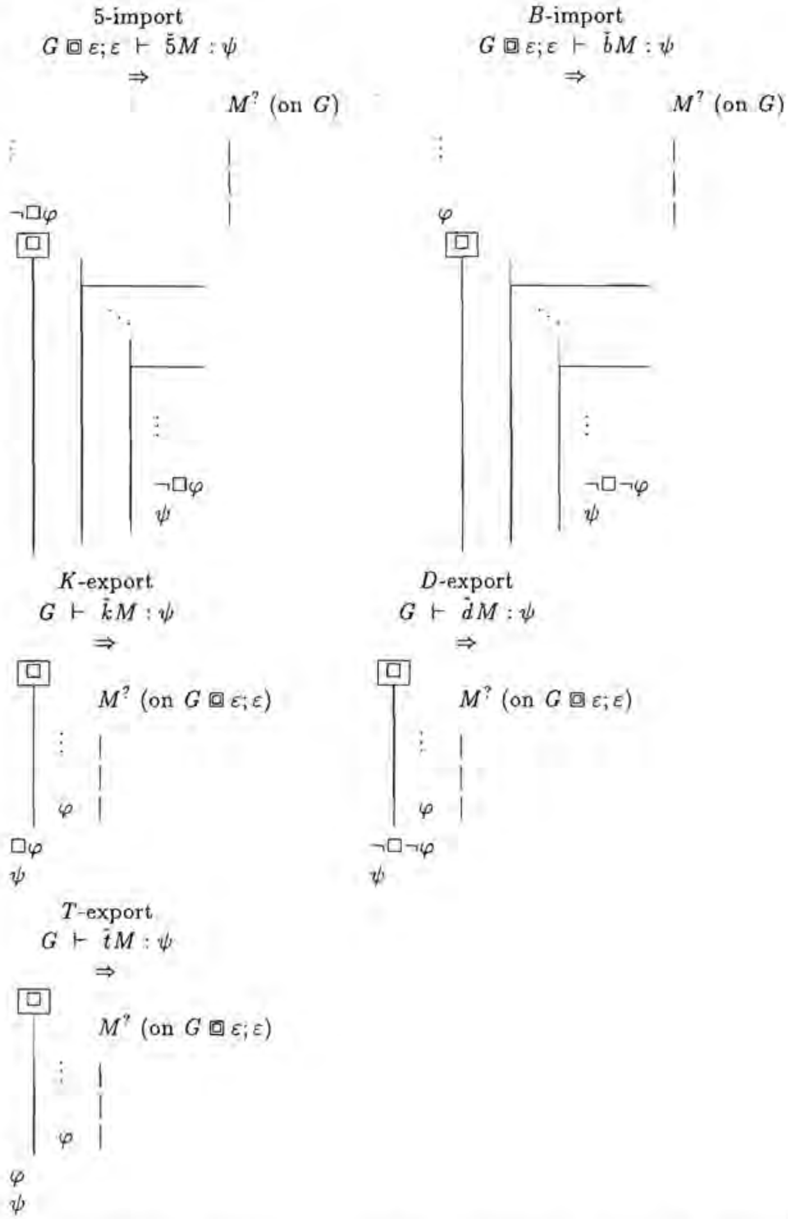
$\forall x \in D\psi$
φ


proof application
G ⊢ MN : φ
⇒
N? (on G)
χ
M? (on G)
χ ⊃ ψ
ψ
φ


object application
G ⊢ Nt : φ
⇒
t? (on F)
t ∈ D
N? (on G)
∀x ∈ D.ψ
ψ[t/x]
φ


K-import
G ⊡ ε; ε ⊢ k̃M : ψ
⇒
M? (on G)
□φ
φ
ψ


4-import
G ⊡ ε; ε ⊢ 4̃M : ψ
⇒
M? (on G)
□φ
□φ
ψ



Given this more general version of '?' it should be possible to prove the soundness of mapping $\Lambda\Box PRED2$-terms to OK-proofs in $\Box PRED2$ in the same way as before.

Compared to the propositional logic, there are no new ways in which the composition of '!' and '?' can transform natural deduction proofs. The same techniques of cleaning etc. that were used for the second order propositional proofs in chapter 2 can be applied to the modal predicate logical deductions. Hence it should be possible to prove a 'back and forth'-equivalence modulo '$=_{doubles}$' for the class of A-OK deductions of $\Box PRED2$ defined as before: a natural deduction proof Σ of φ is *A-OK* iff:

Σ is OK

Σ is clean

Σ has ordered premises for $\forall$-*elim* and $\supset$-*elim*

Σ has no multiple exports

5.1.5. From $\Lambda\Box PRED2$ to $\lambda\Box PRED2$ and back

The last step in mapping modal natural deduction to modal pure type systems is to show that the system $\Lambda\Box PRED2$ is equivalent to the MPTS $\lambda\Box PRED2$. In [Geuvers 1993] the proof of equivalence of the intermediate and the 'target' type system of the interpretation hinges on the following basic property of PTSs in the Logic Cube.

5.1.6. PROPOSITION. *In $\lambda PRED\omega$ we have the following.*
If $\Gamma \vdash M : A$, then $\Gamma_D, \Gamma_T, \Gamma_P \vdash M : A$ where

- *$\Gamma_D, \Gamma_T, \Gamma_P$ is a sound permutation of Γ (it is a legal context that is a permutation of Γ),*
- *Γ_D only contains declarations of the form $x : Set$,*
- *Γ_T only contains declarations of the form $x : A$ with $\Gamma_D \vdash A : Set/Type^p$,*
- *Γ_P only contains (ordered) declarations of the form $x : \varphi$ with $\Gamma_D, \Gamma_T \vdash \varphi : Prop$,*
- *if $A \equiv Set/Type^p$, then $\Gamma_D \vdash M : A$,*
- *if $\Gamma \vdash A : Set/Type^p$, then $\Gamma_D, \Gamma_T \vdash M : A$.*

In the generalized contexts of the MPTS $\lambda\Box PRED2$ we have these properties for each of the constituting contexts $\Gamma_1, \ldots \Gamma_n$ of a generalized context G:

5.1.7. PROPOSITION. *In $\lambda\Box PRED2$ we have the following.*
If $G \vdash M : A$, where $G \equiv \Gamma_1 \boxdot \ldots \boxdot \Gamma_n$, then $\Gamma^1_D, \Gamma^1_T, \Gamma^1_P \boxdot \ldots \boxdot \Gamma^n_D, \Gamma^n_T, \Gamma^n_P \vdash M : A$ where

- *$\Gamma^i_D, \Gamma^i_T, \Gamma^i_P$ is a permutation of Γ^i for all $i : 1 \leq i \leq n$ and $\Gamma^1_D, \Gamma^1_T, \Gamma^1_P \boxdot \ldots \boxdot \Gamma^n_D, \Gamma^n_T, \Gamma^n_P$ is legal,*
- *Γ^i_D only contains declarations of the form $x : Set$,*
- *Γ^i_T only contains declarations of the form $x : A$ with $\Gamma^1_D, \Gamma^1_T, \Gamma^1_P \boxdot \ldots \boxdot \Gamma^i_D \vdash A : Set/Type^p$ for all $i : 1 \leq i \leq n$,*
- *Γ^i_P only contains (ordered) declarations of the form $x : \varphi$ with $\Gamma^1_D, \Gamma^1_T, \Gamma^1_P \boxdot \ldots \boxdot \Gamma^i_D, \Gamma^i_T \vdash \varphi : Prop$,*

- *if* $A \equiv Type^p$, *then* $\Gamma^1_D, \Gamma^1_T, \Gamma^1_P \boxdot \ldots \boxdot \Gamma^i_D \vdash M : A$,
- *if* $\Gamma_1 \boxdot \ldots \boxdot \Gamma_n \vdash A : Set/Type^p$, *then* $\Gamma^1_D, \Gamma^1_T, \Gamma^1_P \boxdot \ldots \boxdot \Gamma^n_D, \Gamma^n_T \vdash M : A$.

Proving equivalence now amounts to showing that we can move back and forth between the 'three part'-contexts $(\Gamma^i_D, \Gamma^i_T, \Gamma^i_P)$ of $\lambda\Box PRED2$ and the 'two part'-contexts (Γ_i, Δ_i) of $\Lambda\Box PRED2$ while preserving derivability. The proof is analogous to that of the equivalence of $\Box PROP2$ and $\lambda\Box PROP2$ given in chapter 3, mapping both the Γ^i_D- and Γ^i_T-parts of $\lambda\Box PRED2$-contexts to the Γ_i-parts of the generalized contexts in $\Lambda\Box PRED2$.

5.2. Modal choices

In the previous sections we have outlined a type theoretical formalism in which modal predicate logical reasoning can be interpreted. Since this system was obtained by combining the standard PTS-interpretation of predicate logic with Fitch-style modal deduction, the question arises what kind of modal predicate logic we have ended up with. To answer this question we will now look at two significant parameters of MPL-systems: the interaction between quantification and modality, and the behaviour of identity.

5.2.1. Quantifier/Box interaction

All possible interactions between the modal operator and the quantifiers are expressed in the following formulas (where $\varphi(x)$ means that x may appear in φ):

1 $\forall x \in D.\Box\varphi(x) \supset \Box\forall x \in D.\varphi(x)$

2 $\Box\forall x \in D.\varphi(x) \supset \forall x \in D.\Box\varphi(x)$

3 $\exists x \in D.\Box\varphi(x) \supset \Box\exists x \in D.\varphi(x)$

4 $\Box\exists x \in D.\varphi(x) \supset \exists x \in D.\Box\varphi(x)$

Formula 1 is known as the *Barcan formula*. Its name derives from Ruth C. Barcan who called attention to it, and it has given rise to some philosophical controversy. Under the standard interpretation the Barcan formula means that if everything necessarily possesses a certain property φ, then it is necessarily the case that everything possesses that property. But one could argue that even if everything that exists is necessarily φ, this does not preclude the possibility that there might be (or might have been) some things which are not φ at all - and in that case it would not be a necessary truth that everything is φ. Not surprisingly Formula 2, the converse of formula 1, is also known as the 'Converse Barcan formula'.

Of the principles involving '$\Box$' and '$\exists$', formula 4 is the controversial one: the fact that there necessarily exists a person who will be prime minister of the country after the next elections ($\Box\exists x \in D.\varphi(x)$), does not imply that there exists a person who necessarily will be prime minister after the next elections ($\exists x \in D.\Box\varphi(x)$). This principle can only be made plausible when dealing with 'intensional objects' such as 'the top card in a pack of cards' used in the description of the rules of a card game, where it is conceived as a single object (even though the top card may be e. g. jack of hearts at one moment in the card game and queen of diamonds the next).

In this section we investigate the interaction of the modal operator and the quantifiers in $\Box PRED2$ and $\lambda\Box PRED2$ by means of the principles stated above, starting from the standard model theoretic account of the Barcan and Converse Barcan formula.

From the point of view of model theory, the validity of the Barcan Formula is connected with the question what is to count as a 'possible world'. Given the picture of a 'world' as a set of objects with various properties and standing in various relations to one another, what states of affairs other than the actual one do we deem possible? We assume that in those worlds objects can have properties and relations different from those in the actual world. But what do we allow with respect to the 'inventory'?

a Worlds containing the same objects as the actual world

b Worlds in which new objects 'appear'

c Worlds from which objects may have 'disappeared'

d Worlds in which objects may have 'appeared' as well as 'disappeared'

Each of these answers can be restated in terms of conditions on the relation between the domain (set of objects) of a world (D_i) and the domains of its accessible worlds (D_j):

a $\forall w_j(w_i R w_j \Rightarrow D_i = D_j)$, 'constant domains'

b $\forall w_j(w_i R w_j \Rightarrow D_i \subseteq D_j)$, 'growing domains'

c $\forall w_j(w_i R w_j \Rightarrow D_j \subseteq D_i)$, 'shrinking domains'

d $\forall w_j(w_i R w_j \Rightarrow D_i \ ?\ D_j)$ 'unrelated domains'

In the cases where domains vary, there is a choice to be made with respect to the valuation of atomic formulas (φ) in a world (w_i) when the formula contains individual-variables which get assigned to objects not present in the domain of that world:

i $V(\varphi, w_i)$ is *undefined* if the value assigned to any of the individual variables is not in D_i. This undefinedness then 'percolates upwards' rendering valuations of more complex formulas referring to 'disappeared individuals' undefined, as can be seen from the valuation clauses for '$\forall$' and $\Box$':

$\forall$ For any wff φ, any individual variable, x, and any world, $w_i \in W$, $V(\forall x\varphi x, w_i) = 1$ iff for every V', which assigns to x any member of D_i and is otherwise the same as V, $V'(\varphi x, w_i) = 1$; and $V(\forall x\varphi x, w_i) = 0$ iff there is some such V' for which $V'(\varphi x, w_i) = 0$. **Otherwise** $V(\forall x\varphi x, w_i)$ undefined.

$\Box$ For any wff φ, and any $w_i \in W$, $V(\Box\varphi, w_i) = 1$ iff for every w_j such that $w_i R w_j$, $V(\varphi, w_j) = 1$; and $V(\Box\varphi) = 0$ iff **for every such** w_j, $V(\varphi, w_j)$ **is defined** and for some such w_j, $V(\varphi, w_j) = 0$ (Thus $V(\Box\varphi, w_i)$ is undefined iff for some such w_j, $V(\varphi, w_j)$ is undefined)

ii $V(\varphi, w_i) = 1$ *or* 0 according as the value-world pair (value assigned by V to the variable(s) in φ, w_i) is in the extension ($V(\varphi)$) of φ or not. In this way every formula has a truth value everywhere. '$\forall$' and '$\Box$' are evaluated applying the clauses as under *i* without the definedness-proviso.

Not all combinations of the answers a, b, c, d and i, ii are well-charted systems of modal logic. Traditionally one would choose a valuation as under i and impose 'growing domains' ($D_i \subseteq D_j$) on the semantics, and hence look only at systems $i\,a$ and $i\,b$. Under valuation i, the Barcan formula holds iff the domains are shrinking and Converse Barcan holds iff the domains are growing. Hence we have the following possibilities for the i-systems:

i a Barcan and Converse Barcan

i b Converse Barcan

i c Barcan

i d -

The only well-known formalism using valuation option ii, is the so-called Kripke's semantics for MPL ([Kripke 1963]). It is a system of type $ii\ d$, where the relation between the domains is unknown, and both Barcan and Converse Barcan are invalid regardless of the modal strength of the system.

The many sorted modal predicate logic $\Box PRED2$ and the MPTS $\lambda\Box PRED2$ offer even stranger possibilities: using n-ary predicates 'multiple Barcan formulas' could be investigated: for instance $\forall x \in A.\forall y \in B.(\Box\varphi(x,y) \supset \Box(\forall x \in A.\forall y \in B.\varphi(x,y))$ in two sorts A and B, where the domains of sort A could be shrinking while those of sort B are growing. However, we will not look into these exotic formulas since it is difficult enough to give a Fitch-style deduction account of the 'one-sorted' Barcan and Converse Barcan formulas.

Converse Barcan

In $\Box PRED2$, the Converse Barcan formula is a theorem. Taking A to be the (basic) domain of the logic, it can be proved as follows:

1.	$\Box\forall x \in A.\varphi(x)$	
2.	$\quad y \in A$	
3.	$\quad \Box\forall x \in A.\varphi(x)$	(reiteration 1)
	$\quad\quad \Box$	
4.	$\quad\quad \forall x \in A.\varphi(x)$	(K-import 3)
5.	$\quad\quad y \in A$	(term)
6.	$\quad\quad \varphi(y)$	($\forall$-elim 4,5)
7.	$\quad \Box\varphi(y)$	(K-export 6)
8.	$\forall y \in A.\Box\varphi(y)$	($\forall$-intro 2-7)
9.	$\Box(\forall x \in A.\varphi(x)) \supset (\forall y \in A.\Box\varphi(y))$	($\supset$-intro 1-8)

The crucial step in this derivation is the use of the term rule in line 6. Since the term rule allows us to write any term $t \in D$ anywhere in a proof, we can eliminate the quantified formula $\forall x \in A.\varphi(x)$ in the modal subordinate proof with the variable $y \in A$ *assumed in the main proof*. This means that the term rule allows us to 'reiterate' term variables across a modal interval, thus circumventing the restriction that reiteration may only take place between two intervals of equal modal depth. In Fitch's original system of modal predicate logical deduction [Fitch 1952] the rules for universal quantification are somewhat different, but Converse Barcan is also a theorem and it is based on a similar implicit 'free reiteration' of variables.

The derivation of Converse Barcan in $\lambda\Box PRED2$ is completely analogous to the natural deduction proof given above, but the MPTS deals with the 'reiteration' of the variable assumption in a more explicit way:

1. $\Gamma, z : \Box(\forall x : A.\varphi \vdash z : \Box(\forall x : A.\varphi(x))$
2. $\Gamma, z : \Box(\forall x : A.\varphi(x)), y : A \vdash z : \Box(\forall x : A.\varphi(x))$
3. $\Gamma, z : \Box(\forall x : A.\varphi(x)), y : A \vdash y : A$
4. $\Gamma, z : \Box(\forall x : A.\varphi(x)), y : A \boxdot \varepsilon \vdash \check{k}z : (\forall x : A.\varphi(x))$ (K-import 2)
5. $\Gamma, z : \Box(\forall x : A.\varphi(x)), y : A \boxdot \varepsilon \vdash y : A$ (transfer$_3$ 3)
6. $\Gamma, z : \Box(\forall x : A.\varphi(x)), y : A \boxdot \varepsilon \vdash (\check{k}z)y : \varphi(y)$
7. $\Gamma, z : \Box(\forall x : A.\varphi(x)), y : A \vdash \hat{k}((\check{k}z)y) : \Box\varphi(y)$ (K-export 7)
8. $\Gamma, z : \Box(\forall x : A.\varphi(x)) \vdash \lambda y : A.(\hat{k}((\check{k}z)y)) : (\forall y : A.\Box\varphi(y))$
9. $\Gamma \vdash \lambda z : \Box(\forall x : A.\varphi(x)).\lambda y : A.(\hat{k}((\check{k}z)y)) : (\Box\forall x : A.(\varphi x)) \supset \forall y : A.(\Box\varphi(y))$

In this proof the use of the assumption $y : A$ (line 3) in the subordinate context (line 5) is motivated by an application of the *transfer*$_3$ rule. Intuitively using the *transfer*$_3$-rule to reiterate a set-variable in a modal subcontext corresponds to having some sort of domain inclusion requirement: any variable ranging over a set(-type) in the main context will be a variable ranging over that set in the subordinate context.

The *transfer*$_3$-rule is not only sufficient for the derivation of Converse Barcan, it also necessary. This can be seen from the following attempt to derive the formula without *transfer*$_3$. It starts out (line 1-4) as above, but we can no longer eliminate the universal quantifier in $\check{k}z : (\forall x : A.(Px))$ with $y : A$ from line 3. There is no way to derive $y : A$ in the subordinate context from the fact that $y : A$ is derivable in the main context.

1. $\Gamma, z : \Box(\forall x : A.(Px)) \vdash z : \Box(\forall x : A.(Px))$ ($\Gamma \vdash A : Set$)
2. $\Gamma, z : \Box(\forall x : A.(Px)), y : A \vdash z : \Box(\forall x : A.(Px))$ (weakening)
3. $\Gamma, z : \Box(\forall x : A.(Px)), y : A \vdash y : A$
4. $\Gamma, z : \Box(\forall x : A.(Px)), y : A \boxdot \varepsilon \vdash \check{k}z : (\forall x : A.(Px))$ (K-import 2)
5. $\Gamma, z : \Box(\forall x : A.(Px)), y : A \boxdot \varepsilon \vdash A : Set$ (transfer$_2$)
6. $\Gamma, z : \Box(\forall x : A.(Px)), y : A \boxdot u : A \vdash \check{k}z : (\forall x : A.(Px))$ (weakening 4,5)
7. $\Gamma, z : \Box(\forall x : A.(Px)), y : A \boxdot u : A \vdash u : A$ (start 5)
8. $\Gamma, z : \Box(\forall x : A.(Px)), y : A \boxdot u : A \vdash (\check{k}z)u : Pu$

The quantifier can only be eliminated by means of a variable of type A *created in the subordinate context* (line (5)-(7)). But the creation of a variable $u : A$ automatically leads to the creation of the assumption $u : A$ in the subordinate context. Hence this procedure will never lead to a conclusion of the subordinate derivation with a proofterm of the form $\Box Pu$ (let alone $\Box Py$), since the assumption $u : A$ would have to be discharged from the subordinate context *before* the K-export rule (or any other export rule) can be applied. Without the *transfer*$_3$-rule, variables are 'confined' to that part of a generalized context in which they are introduced (in this case Γ). The interaction between the type theoretical treatment of variables and the modal deduction requirements of the export rule blocks the derivation of Converse Barcan.

Since all meta theoretical properties of MPTSs proved in the chapter 3 can be shown to hold for MPTS's *without* $transfer_3$, we have a choice between $\lambda\Box PRED2$ without $transfer_3$ where Converse Barcan is not a theorem, and $\lambda\Box PRED2$ with $transfer_3$ where Converse Barcan is a theorem. We will discuss this choice along with other, forthcoming, 'modal choices' at the end of this chapter.

Barcan

The Barcan formula is not derivable in $\Box PRED2$ and in Fitch's system ([Fitch 1952]). The reason for this is somewhat obscure. Fitch contends himself with the following remark: 'It is not possible to derive $\Box(x)\varphi x$ from $(x)\Box\varphi$ unless a special rule to that effect is assumed. Such a rule would not violate the consistency of the system, and it would seem to be a valid principle'. Similarly, Fitting gives matching natural deduction- and tableau systems for various modal logics throughout [Fitting 1983], but without further explanation he resorts to tableau systems only for modal predicate logics with varying domains.

The discussion of Converse Barcan may have created the impression that there is an analogy between 'free iteration' in Fitch-style deduction and domain inclusion in model theory. However, things are not that simple. The Barcan formula does not become derivable when we allow variable assumptions of a modal subordinate proof to be used in the main proof:

1.	$\forall x \in A.\Box\varphi(x)$	
2.	$y \in A$	('backward free reiteration' 4)
3.	$\Box\varphi(y)$	($\forall$-elim 1,2)
	$\Box$	
4.	$y \in A$	
5.	$\varphi(y)$	(K-import 3)
6.	$\forall y \in A.\varphi(y)$	($\forall$-intro 4-5)
7.	$\Box\forall y \in A.\varphi(y)$	(K-export 6)
8.	$(\forall x \in A.\Box\varphi(x)) \supset (\Box\forall y \in A.\varphi(y))$	($\supset$-intro 1-7)

The free reiteration has to work 'backwards' (from line 4 to line 2) in the proof figure to prove Barcan (note that the occurrence of $y \in A$ in line 2 cannot be an application of term, since in that case we would have had to choose a variable different from y for the assumption in line 4).

Apparently the Barcan Formula is not derivable, even if we allow free reiteration of variable assumptions. A reason for this is that there is a fundamental asymmetry between subordinate proofs that introduce quantifiers and modal subordinate proofs. In order to bring out this asymmetry, we use the $\forall$-quantor in the original formulation of Fitch ([Fitch 1952]):

(u q elim)	(u q int)
$\vdots$	$\vdots$
$\forall x(\ldots x \ldots)$	$\boxed{x}$
$(\ldots a \ldots)$ (a is a term)	$\quad\vdots$
	$\quad(\ldots x \ldots)$
	$\forall x(\ldots x \ldots)$

The elimination rule (u q elim) is the same as before, the occurrences of x in the expression $(\ldots x \ldots)$ are replaced by occurrences of the term a. The difference is in the introduction rule: $\forall x(\ldots x \ldots)$ is a consequence of a categorical subordinate proof (all assumptions discharged) that is *general with respect to* x and has $(\ldots x \ldots)$ as an item. Like in the case of the modal operator the introduction of a universal quantifier requires a categorical subordinate proof with a restricted rule of reiteration: *general with respect to* x means that only formulas from the main proof which do not contain free occurrences of x may be repeated in an x-general subordinate proof.

To understand the difference between Fitch-style derivations of Converse Barcan and Barcan we look at them as if they were derivations in predicate logic. For this purpose the $\Box$ is interpreted as a universal quantification over the set of (accessible) worlds (W), and hence modal subordinate proofs are treated as 'w-general' proofs.

Converse Barcan	Barcan
$\forall w \in W.\forall x \in A.\varphi(x,w)$	$\forall x \in A.\forall w \in W.\varphi(x,w)$
$\quad\boxed{x}$	$\quad\boxed{w}$
$\quad\forall w \in W.\forall x \in A.\varphi(x,w)$	$\quad\forall x \in A.\forall w \in W.\varphi(x,w)$
$\qquad\boxed{w}$	$\qquad\boxed{x}$
$\qquad\forall w \in W.\forall x \in A.\varphi(x,w)$	$\qquad\forall x \in A.\forall w \in W.\varphi(x,w)$
$\qquad\forall x \in A.\varphi(x,w)$	$\qquad\forall w \in W.\varphi(x,w)$
$\qquad\varphi(x,w)$	$\qquad\varphi(x,w)$
$\quad\forall w \in W.\varphi(x,w)$	$\quad\forall x \in A.\varphi(x,w)$
$\forall x \in A.\forall w \in W.\varphi(x,w)$	$\forall w \in W.\forall x \in A.\varphi(x,w)$

In this format proving Barcan and Converse Barcan boils down to 'exchanging $\forall$-quantifiers', yet the left proof is allowed and the right proof is not. Both the w-general and x-general proof are categorical, the difference lies in the restriction on the iteration of formulas into these proofs:

> **x-general proofs**: a formula (ψ) may be reiterated inside an x-general proof if x does not occur free in it ($x \notin FV(\psi)$).
>
> **w-general proofs**: a formula ψ may be reiterated inside a w-general proof if it is of the form $\forall w \in W\psi'$.

The first restriction concerns the *occurrence of free variables* in a formula, the second concerns the *general form* of the formula. As a consequence closed formulas with $\forall w \in W$ as their main connective can always be reiterated in 'quantifier-introducing' subordinate proofs, but the reiteration of formulas with $\forall x \in A$ as main connective in modal subordinate proofs is not allowed. This asymmetry blocks proofs of Barcan, but does not affect proofs of its converse: the proof of Converse Barcan can go through since we are allowed to reiterate a formula of the general form $\forall w \in W.\varphi$ inside an x-general proof (line 2) as long as x does not occur free in it (quod non). The proof of Barcan does not go through since we are not allowed to reiterate a formula of the form $\forall x \in A.\varphi$ in a w-general proof (line 2).

This excursion to a predicate logic does not completely explain why it is impossible to give a Fitch-style deduction rule that makes Barcan a theorem. Judging by the derivations above, we could obtain both Barcan and Converse Barcan by changing the reiteration condition on w-general proofs to:

> **w-general proofs**: a formula may be reiterated inside a w-general proof if w does not occur free in it.

However, it is completely unclear what Fitch-style import rule corresponds to this condition.

Even though free iteration of variable assumptions does not make the Barcan formula derivable in $\Box PRED2$, we may fare better in $\lambda\Box PRED2$ since it seems to offer ways to circumvent the strict nesting constraints of Fitch-style natural deduction. Therefore we add a type theoretical rule for 'backward free reiteration':

$$varuse\ \frac{G \boxdot \Gamma \ \vdash\ y : A : Set}{G \ \vdash\ y : A}$$

In this way the set variables from the subordinate context can be used (but not abstracted over) in the main context. This rule looks a lot like the derivable rule $refsnart_3$ (section 3.3),

$$refsnart_3\ \frac{G \boxdot \varepsilon \ \vdash\ y : A : Set}{G \ \vdash\ y : A}$$

but the vital difference is that $refsnart_3$ demands that the subordinate context is empty. This blocks the possibility of bringing set-variables assumed in the subordinate context back to the main context.

With the *varuse* rule the Barcan formula can be derived similarly to the way it would be proved using indexed tableaux for MPL with shrinking domains[1]:

1 Start a subordinate context and 'create' the variable y (line 1-3).

2 'Take a break' from that subordinate proof, go back to the main context and use the y from the subordinate context (by the *varuse*-rule) to eliminate the quantifier (line 4-6).

3 Return to the subcontext with the modal formula just derived on the main context (7-9).

[1]Cf. [Fitting 1983].

1.	$\Gamma, z : \forall x : A.\Box(\varphi x) \vdash A : Set$	$(\Gamma \vdash A : Set)$
2.	$\Gamma, z : \forall x : A.\Box(\varphi x) ⧈ \varepsilon \vdash A : Set$	(transfer$_1$ 1)
3.	$\Gamma, z : \forall x : A.\Box(\varphi x) ⧈ y : A \vdash y : A$	(start 2)
4.	$\Gamma, z : \forall x : A.\Box(\varphi x) \vdash y : A$	(varuse 3)
5.	$\Gamma, z : \forall x : A.\Box(\varphi x) \vdash z : \forall x : A.\Box(\varphi x)$	
6.	$\Gamma, z : \forall x : A.\Box(\varphi x) \vdash zy : \Box(\varphi y)$	
7.	$\Gamma, z : \forall x : A.\Box(\varphi x) ⧈ \varepsilon \vdash \check{k}(zy) : \varphi y$	(K-import 6)
8.	$\Gamma, z : \forall x : A.\Box(\varphi x) ⧈ y : A \vdash y : A$	
9.	$\Gamma, z : \forall x : A.\Box(\varphi x) ⧈ y : A \vdash \check{k}(zy) : \varphi y$	
10.	$\Gamma, z : \forall x : A.\Box(\varphi x) ⧈ \vdash \lambda y : A.(\check{k}(zy)) : \forall y : A.\varphi y$	
11.	$\Gamma, z : \forall x : A.\Box(\varphi x) \vdash \hat{k}(\lambda y : A.(\check{k}(zy))) : \Box\forall y : A.\varphi y$	(K-export 10)
12.	$\Gamma \vdash \lambda z : \forall x : A.\Box(\varphi x).(\hat{k}(\lambda y : A.(\check{k}(zy)))) : (\forall x : A.\Box(\varphi x)) \supset \Box(\forall y : A.\varphi y)$	

Obviously 'taking a break' from a modal subordinate proof is impossible in natural deduction proofs. The fact that '?' maps the proof term for Barcan to the 'backwards' natural deduction proof above confirms that the *varuse*-rule takes us out of the Fitch-style framework. That adding *varuse* also takes us outside of the 'MPTS-framework' does not become apparent until one tries to prove that the meta theoretical properties of $\lambda\Box PRED2$ are preserved under the addition.

Already the basic lemmas from the section preliminaries of chapter 3 do not hold. The Free Variable Lemma (ii) says that for legal contexts G, if $G \vdash A : B$, then $FV(A), FV(B) \subseteq FV(G)$. In line 4 $y : A$ is derivable on context Γ, but $FV(y) \not\subseteq FV(\Gamma)$, since y was introduced *as a fresh variable* by the application of *Start* in line 3 $FV(y) \not\subseteq FV(\Gamma ⧈ \varepsilon)$. Hence adding *varuse* is a too expensive way to make the Barcan Formula a theorem of $\lambda\Box PRED2$: we lose the Free Variable Lemma and thereby most of the meta theoretical properties proven earlier for the Modal Logic Cube.

Another obvious possibility is to add a rule that mirrors *transfer*$_3$ in another way:

$$varmove\ \frac{G, x : A ⧈ \varepsilon \vdash B : C \qquad G \vdash A : Set}{G ⧈ x : A \vdash B : C}$$

This rule allows that the *discharge* of a variable created in the main context takes place in the subordinate context, as opposed to *transfer*$_3$ that allows a variable created in the main context to be *used* in a subordinate context.

For the cases in which B is not a proof, *varmove* is already a derived rule of $\lambda\Box PRED2$ since it can be viewed as a combination of Block insertion and Block deletion (cf. chapter 3):

1.	$G, x : A ⧈ \varepsilon \vdash B : C$	
2.	$G, x : A \vdash B : C$	(Block deletion)
3.	$G ⧈ x : A \vdash B : C$	(Block insertion)

However, for deriving Barcan we typically need B to be a proof and so *varmove* does add something new. Using *varmove*, Barcan can be derived as follows:

1. $\Gamma, z : (\forall x : A.(\Box(Px))) \vdash z : \forall x : A.(\Box(Px))$ $(\Gamma \vdash A : Set)$
2. $\Gamma, z : (\forall x : A.(\Box(Px))), y : A \vdash y : A$
3. $\Gamma, z : (\forall x : A.(\Box(Px))), y : A \vdash zy : \Box(Py)$
4. $\Gamma, z : (\forall x : A.(\Box(Px))), y : A \boxdot \varepsilon \vdash \check{k}(zy) : Py$ (K-import 3)
5. $\Gamma, z : (\forall x : A.(\Box(Px))) \boxdot y : A \vdash \check{k}(zy) : Py$ (varmove 4)
6. $\Gamma, z : (\forall x : A.(\Box(Px))) \boxdot \varepsilon \vdash \lambda y : A.(\check{k}(zy)) : (\forall y : A.Py)$
7. $\Gamma, z : (\forall x : A.(\Box(Px))) \vdash \hat{k}(\lambda y : A.(\check{k}(zy))) : \Box(\forall y : A.Py)$ (K-export 6)
8. $\Gamma \vdash \lambda z : (\forall x : A.(\Box(Px))).(\hat{k}(\lambda y : A.(\check{k}(zy)))) : (\forall x : A.\Box(Px)) \supset \Box(\forall y : A.Py)$

Again it is immediately clear that the extra rule takes us outside the Fitch-style framework, '?' maps the proofterm for Barcan to the 'backward' natural deduction proof from the beginning of this section.

Unfortunately adding the *varmove*-rule also leads to problems in proving the basic lemma of the meta theory of MPTSs . In the proof of the Start-lemma (and elsewhere), the induction on the lenght of the derivation for the export cases uses the fact that if $G \boxdot \Gamma \vdash A : B$, there must have been a derivable statement $C : D$ such that $G \vdash C : D$, since all derivations start from the empty context (ε) and at some point in the derivation the final $\boxdot$ must have been introduced by a rule leading from G to $G \boxdot \varepsilon$ for the first time. If we add *varmove*, this inference need no longer hold. Looking at lines 5 (or 6) above, it should then be possible to move upwards in the derivation to find the place where $\Gamma, z : (\forall x : A.(\Box(Px)))$ is turned into $\Gamma, z : (\forall x : A.(\Box(Px))) \boxdot \varepsilon$ for the first time, but there is no such line in the entire derivation!

Although this discussion of the Barcan formula is by no means conclusive, there seems to be no intuitive rule which makes Barcan a theorem, neither in $\Box PRED2$ nor in $\lambda\Box PRED2$. Apparently the analogy between Fitch-style deduction and Barendregt-style type theoretical derivation is strong enough to make $\lambda\Box PRED2$ inherit the impossibility of deriving Barcan from $\Box PRED2$. Unfortunately the reasons for this impossibility are still not completely understood. Hence, if the Barcan formula is to be a theorem of $\lambda\Box PRED2$, there is nothing we can do but follow Fitch's advise and add it as a logical axiom.

Existential interaction

With respect to the interaction between '$\Box$' and '$\exists$', the situation is very similar to that for Barcan and Converse Barcan. The 'harmless' principle $\exists x \in D.\Box\varphi(x) \supset \Box\exists x \in D.\varphi(x)$ is a theorem of $\Box PRED2$ and of Fitch's original system ([Fitch 1952]), whereas the disputed $\Box\exists x \in D.\varphi(x) \supset \exists x \in D.\Box\varphi(x)$ is not derivable and can only be added as an axiom.

Like for Converse Barcan, the derivation of $\exists x \in D.\Box\varphi(x) \supset \Box\exists x \in D.\varphi(x)$ in $\Box PRED2$ depends on 'free reiteration' of variable assumptions. Due to the second order definition of the existential quantifier in $\Box PRED2$, we prove that $\exists x \in D.\Box\varphi(x) \supset \Box\exists x \in D.\varphi(x)$ is a theorem by giving a natural deduction proof for $(\forall\alpha \in Prop.(\forall x \in A.\Box\varphi \supset \alpha) \supset \alpha) \supset \Box(\forall\beta \in Prop.(\forall y \in A.\varphi(y) \supset \beta) \supset \beta)$:

1.	$\forall\alpha \in \mathit{Prop}.(\forall x \in A.\Box\varphi \supset \alpha) \supset \alpha$	
2.	$\Box(\forall\beta \in \mathit{Prop}.(\forall y \in A.\varphi(y) \supset \beta) \supset \beta) \in \mathit{Prop}$	(term)
3.	$\forall x \in A.(\Box\varphi(x) \supset (\Box(\forall\beta \in \mathit{Prop}.(\forall y \in A.\varphi(y) \supset \beta) \supset \beta))) \supset$ $(\Box(\forall\beta \in \mathit{Prop}.(\forall y \in A.\varphi(y) \supset \beta) \supset \beta))$	
4.	$x \in A$	
5.	$\Box\varphi(x)$	
	$\boxed{\Box}$	
6.	$\beta \in \mathit{Prop}$	
7.	$(\forall y \in A.\varphi(y) \supset \beta)$	
8.	$x \in A$	(free reiteration 4)
9.	$\varphi(x) \supset \beta$	
10.	$\varphi(x)$	(K-import 5)
11.	β	
12.	$(\forall y \in A.\varphi(y) \supset \beta) \supset \beta$	
13.	$\forall\beta \in \mathit{Prop}.(\forall y \in A.\varphi(y) \supset \beta) \supset \beta$	
14.	$\Box(\forall\beta \in \mathit{Prop}.(\forall y \in A.\varphi(y) \supset \beta) \supset \beta$	
15.	$\Box\varphi(x) \supset (\Box(\forall\beta \in \mathit{Prop}.(\forall y \in A.\varphi(y) \supset \beta) \supset \beta))$	
16.	$\forall x \in A.(\Box\varphi(x) \supset (\Box(\forall\beta \in \mathit{Prop}.(\forall y \in A.\varphi(y) \supset \beta) \supset \beta)))$	
17.	$\Box(\forall\beta \in \mathit{Prop}.(\forall y \in A.\varphi(y) \supset \beta) \supset \beta)$	
18.	$\forall\alpha \in \mathit{Prop}.(\forall x \in A.\Box\varphi \supset \alpha) \supset \alpha \supset \Box(\forall\beta \in \mathit{Prop}.(\forall y \in A.\varphi(y) \supset \beta) \supset \beta)$	

The idea of the proof is to substitute the consequent of the desired conclusion (line 2) for the α in the assumed antecedent $\forall\alpha \in \mathit{Prop}.(\forall x \in A.\Box\varphi \supset \alpha) \supset \alpha$ (line 1) and then derive $\forall x \in A.(\Box\varphi(x) \supset (\Box(\forall\beta \in \mathit{Prop}.(\forall y \in A.\varphi(y) \supset \beta) \supset \beta)))$, the antecedent of the implication in line 3. The use of free reiteration (line 8) is crucial for proving $\exists x \in D.\Box\varphi(x) \supset \Box\exists x \in D.\varphi(x)$, consequently it can only be derived in $\lambda\Box PRED2$ *with transfer*$_3$.

The formula $\Box\exists x \in D.\varphi(x) \supset \exists x \in D.\Box\varphi(x)$ is even less derivable in $\Box PRED2$ than Barcan, it cannot even be derived if 'backward free reiteration' is allowed. Hence we close the discussion with the remark that if desired it can be added to $\lambda\Box PRED2$ as a logical axiom.

5.2.2. Identity

Predicate logic is often augmented with a primitive dyadic predicate constant '=', representing identity between individuals: '$x = y$' is to mean that x is the same individual as (identical with) y. That two individuals are different can be expressed by $\neg(x = y)$, usually abbreviated as $x \neq y$.

Although the identity predicate looks rather harmless, its properties in combination with the modal operator have given rise to considerable discussion. If individuals are interpreted as particular objects in the domain (the same in every possible world), the following formula becomes valid:

Necessity of Identity: $a = b \supset \Box(a = b)$

This a controversial principal since it claims that identities which are at first sight accidental, like that of Hespherus and Phosphorus, are actually necessary. One option is to accept this conclusion as a necessary a posteriori proposition as Kripke did ([Kripke 1972]); to construe it as a necessary self-identity. An alternative is to interpret individuals in such a way that they may denote different objects in different worlds.

In this section we show that Necessity of Identity holds for $\lambda\Box PRED2$, but also that it does not hold necessarily. In typed λ-calculus, individuals are variables in set types: $x : A$, where $A : Set$. To increase legibility of the following discussion, we will distinguish between the various roles played by variables in the MPTS by using a and b for individual variables, and P, Q, R for predicate variables.

Necessary identity

Necessity of Identity is a theorem of Fitch's original natural deduction system for modal predicate logic ([Fitch 1952]). It is based on very straightforward rules for the introduction and elimination of identity. Identities are introduced in the same way as axioms or terms: for any individual a, $a = a$ may be written without further justification at any stage of a proof. The elimination of identity is by substitution: if an interval in a proof figure contains $a = b$ and a formula in which a occurs, $(\ldots a \ldots)$, then $(\ldots b \ldots)$ may be written as an item of that interval.

=-intro	=-elim	Necessity of Identity
$a = a$	$\vdots$ $a = b$ $\vdots$ $(\ldots a \ldots)$ $(\ldots b \ldots)$	1. $a = b$ 2. $\Box$ $a = a$ (=-intro) 3. $\Box(a = a)$ 4. $\Box(a = b)$ (=-elim) 5. $(a = b) \supset \Box(a = b)$

As shown above, these rules allow a simple proof of $a = b \supset \Box(a = b)$ in any normal modal logic.

Like PTSs, MPTSs do not contain a primitive notion of identity. But since $\lambda\Box PRED2$ is a second order system, a form of 'Leibniz identity' can be expressed in its language[2] due to the rule $(Type^p, Prop, Prop)$ which makes quantification over predicates possible. We shall say that a is identical to b iff every predicate P which holds for a also holds for b:

$$\text{`='} \qquad \lambda x : A.(\lambda y : A.(\forall P : A \to Prop.(Px \to Py))) : (A \to (A \to Prop))$$

Since individuals have set types, a and b will be identical if they are of members of the same set ($A : Set$), that satisfy the same predicates over that set. Note that in a many-sorted logic a definition of Leibniz identity would be needed for each of the domains of individuals, the expression above only defines identity with respect to the set A. The proposition that a and b are Leibniz-identical is obtained from '=' by applying $a : A$ and $b : A$ to it (and using β-conversion):

[2]This alternative is also mentioned in [Hughes and Cresswell 1972].

'$a = b$' $\quad (\forall P : A \rightarrow Prop.(Pa \rightarrow Pb)) : Prop$

If Leibniz Identity is added to $\lambda\Box PRED2$ as the definition of identity, all properties that are usually attributed to identity: reflexivity, transitivity, and symmetry, can be derived using the rules of the MPTS. The modal behaviour of the defined equality is very much like that of the primitive equality in natural deduction, we can prove Necessity of Identity via the Necessity of Self-Identity.

1.	$\Gamma \vdash a : A$	$(\Gamma \vdash A : Set)$
2.	$\Gamma \boxdot \varepsilon \vdash a : A$	(transfer$_3$)
3.	$\Gamma \boxdot \varepsilon \vdash A : Set$	(transfer$_1$)
4.	$\Gamma \boxdot \varepsilon \vdash Prop : Type^p$	(type axiom)
5.	$\Gamma \boxdot x : A \vdash Prop : Type^p$	(weakening 3,4)
6.	$\Gamma \boxdot \varepsilon \vdash A \rightarrow Prop : Type^p$	(product 4,5)
7.	$\Gamma \boxdot P : A \rightarrow Prop \vdash a : A$	(weakening 2,6)
8.	$\Gamma \boxdot P : A \rightarrow Prop \vdash P : A \rightarrow Prop$	
9.	$\Gamma \boxdot P : A \rightarrow Prop \vdash Pa : Prop$	
10.	$\Gamma \boxdot P : A \rightarrow Prop, y : Pa \vdash y : Pa$	
11.	$\Gamma \boxdot P : A \rightarrow Prop \vdash \lambda y : Pa.y : (Pa \rightarrow Pa)$	
12.	$\Gamma \boxdot \varepsilon \vdash \lambda P : A \rightarrow Prop.(\lambda y : Pa.y) : (\forall P : A \rightarrow Prop.(Pa \rightarrow Pa))$	
13.	$\Gamma \vdash \hat{k}(\lambda P : A \rightarrow Prop.(\lambda y : Pa.y)) : \Box(\forall P : A \rightarrow Prop.(Pa \rightarrow Pa))$	

Necessity of Self-Identity is proved by first constructing a proof term for $a = a$ on an empty subordinate context, and then exporting it. The derivation in lines 2-12 shows that the introduction rule for '=' in natural deduction is a derived rule in the presence of Leibniz Identity, when $A : Set$ and $a : A$ are present in the context. The presence of $a : A$ in the subordinate context depends on *transfer*$_3$, in a system without this rule $a : A$ would be an assumption of the subordinate context and we could at best show that $\hat{k}(\lambda a : A.(\lambda P : A \rightarrow Prop.(\lambda y : Pa.y)))$ is a proof term for $\Box(\forall a : A.(\forall P : A \rightarrow Prop.(Pa \rightarrow Pa)))$.

The second step in proving necessity of identity for K, substitution by =-elim, also involves more work than its natural deduction counterpart. Starting from the proof object for $\Box(a = a)$ as found in line 13 of the above derivation (which we abbreviate by M in the following) and the assumption $z : (a = b)$, a proofterm for $\Box(\forall P : A \rightarrow Prop.Pa \rightarrow Pb)$ has to be found. Using the fact that we are in a *second order* predicate logic, we define the predicate 'being necessarily identical to a': $\Box(a = x)$. Starting from this predicate $Q =_{def} \lambda x : A.\Box(\forall P : A \rightarrow Prop.Pa \rightarrow Px) : Type^p$, $Qb =_\beta \Box(\forall P : A \rightarrow Prop.Pa \rightarrow Pb)$ can be derived from Qa (self-identity of a) and the assumption that $a = b$:

1.	$\Gamma, z : (a = b) \vdash Q : A \rightarrow Prop$	$(Q \equiv \lambda x : A\Box(\forall P : A \rightarrow Prop.Pa \rightarrow Px))$
2.	$\Gamma, z : (a = b) \vdash M : Qa$	$(Qa =_\beta \Box(\forall P : A \rightarrow Prop.Pa \rightarrow Pa))$
3.	$\Gamma, z : (a = b) \vdash z : (a = b)$	$(a = b) =_{def} (\forall P : A \rightarrow Prop.Pa \rightarrow Pb)$
4.	$\Gamma, z : (a = b) \vdash zQ : (Qa \rightarrow Qb)$	
5.	$\Gamma, z : (a = b) \vdash (zQ)M : Qb$	$(Qb =_\beta \Box(\forall P : A \rightarrow Prop.Pa \rightarrow Pb))$
6.	$\Gamma, z : (a = b) \vdash (zQ)M : \Box(a = b)$	$(\Box(\forall P : A \rightarrow Prop.Pa \rightarrow Pb) =_{def} \Box(a = b))$

Given this proof of $(a = b) \supset \Box(a = b)$, we can also prove the necessity of inequality $(a \neq b) \supset \Box(a \neq b)$ in all systems that are extensions of KB. Since $(a = b) \supset \Box(a = b)$ is a theorem of K, we have by contraposition that $\neg\Box(a = b) \supset \neg(a = b)$ is a theorem and hence by Necessitation that $\Box(\neg\Box(a = b) \supset \neg(a = b))$ is a theorem. The natural deduction proof now runs as follows:

1.	$\Box(\neg\Box(a = b) \supset \neg(a = b))$	(theorem)
2.	$\quad \neg(a = b)$	
3.	$\quad\quad \Box\neg\Box(a = b)$	
	$\quad\quad\quad \boxed{\Box}$	
4.	$\quad\quad\quad \neg\Box(a = b) \supset \neg(a = b)$	(K-import 1)
5.	$\quad\quad\quad \neg\Box(a = b)$	(K-import 3)
6.	$\quad\quad\quad \neg(a = b)$	
7.	$\quad\quad \Box\neg(a = b)$	(K-export 5)
8.	$\quad \Box\neg\Box(a = b) \supset \Box\neg(a = b)$	
	$\quad\quad \boxed{\Box}$	
9.	$\quad\quad \neg\Box(a = b)$	(B-import ($\neg\neg$-elim) 2)
10.	$\quad \Box\neg\Box(a = b)$	(K-export 9)
11.	$\quad \Box\neg(a = b)$	
12.	$\neg(a = b) \supset \Box\neg(a = b)$	

And hence $(a \neq b) \supset \Box(a \neq b)$. If we take N to be the proof object for the theorem in line 1, the corresponding $\lambda\Box PRED2$-term looks like this: $\lambda z : \neg(a = b).((\lambda y : \Box(\neg\Box(a = b).\hat{k}((\check{k}N)(\check{k}y)))(\check{k}(\check{b}z))$.

Contingent identity

Although Necessity of Identity is reasonable from the point of view of Kripke's theory of rigid designation, it can be less reasonable for interpretations of '$\Box$' other than the 'ontological' one of alethic modal logic. For epistemic interpretations of the modal operator, especially the doxastic reading ('belief'), substitution in opaque contexts characteristically fails:

(1) The Babylonians believed that Hesperus is Hesperus.

(2) Hesperus is Phosphorus.

(3) The Babylonians believed that Hesperus is Phosphorus.

From (1) and (2) we cannot correctly infer (3). For this reason, modal predicate logics have been proposed where identity is not necessary, the so-called 'contingent identity logics'. Below we investigate how a type theoretical system with contingent identity can be obtained.

Note that the possibility to form a 'modal' (second order) predicate Q and to substitute it for P in the equality $a = b$ are vital for the derivation of Necessity of Identity. This observation suggests two ways to prohibit that Necessity of Identity becomes a theorem:

> Restrict the formation of predicates in such a way that modal predicates can no longer be formed.
>
> Weaken the notion of Leibniz Identity in such a way that it no longer allows substitution in modal predicates.

The first solution is not very attractive: besides the technical difficulties accompanying the introduction of such restrictions into the standard format of MPTSs, it immediately leads to the undesirable consequence that not even self-identity is derivable anymore. The second solution corresponds to a more traditional move, namely restricting substitution to 'non-opaque contexts'.

As we saw earlier, substitutions of individuals are 'calculated' under Leibniz Identity rather than 'performed'. Given individual a with property Q, the conclusion that the identical individual b has property Q is reached via a predicate logical inference: the universal quantification over predicates P is eliminated with Q and then Qb is inferred from Qa and $Qa \supset Qb$. Since Leibniz Identity quantifies over all predicates P, this works for arbitrary properties. We can block this inference for 'modal' properties by defining a weaker notion of Leibniz Identity:

Weak Leibniz Identity: '$a = b$' $\quad (\forall P : A \rightarrow Prop.(Pa \rightarrow Pb)) : Prop$,
where P ranges over predicates over A *in which* $\Box$ *does not occur.*

Using this identity Qb cannot be inferred from Qa and $a = b$, since the condition on predicates P forbids that the universal quantification over P is eliminated with the predicate Q in which $\Box$ occurs.

Weak Leibniz Identity only guarantees 'identity in a world'; two individuals are identical if they are indiscernible by non-modal predicates. This leaves open the question whether the individuals have different properties in other worlds and hence allows the possibility of contingent identity. Moving to stronger modal systems will not help to restore Necessity of Identity as a theorem, under Weak Leibniz Identity one can at best infer the (necessity of the) possibility of an identity: $a = b \supset \neg\Box\neg(a = b)$ in KT, and $a = b \supset \Box\neg\Box\neg(a = b)$ in KTB.

5.2.3. Concluding remarks

The MPTS $\lambda\Box PRED2$ offers quite a few options known from the model theory of MPL:

$\lambda\Box PRED2$ without *transfer*$_3$:
None of the interaction formulas is derivable, no identity is necessary.

$\lambda\Box PRED2$ with *transfer*$_3$:
Converse Barcan and $\exists x \in D.\Box\varphi \supset \Box\exists x \in D.\varphi$ are theorems.
Other interaction or identity theorems can be obtained by adding:

axioms: for Barcan and for $\Box\exists x \in D.\varphi \supset \exists x \in D.\Box\varphi$

definitions of equality: Weak Leibniz Identity for Necessity of Self-Identity,
Leibniz Identity for Necessity of Identity

From this summary the central role of the *transfer*$_3$ rule immediately becomes apparent. If we think of subordinate contexts as arbitrary accessible worlds, this rule relates the 'inventory' (inhabitants of set-types, individuals) of a world with that of the worlds accessible to it: a variable ranging over the domain of the main context is also a variable ranging over the domain of the subordinate context. In this sense the *transfer*$_3$ rule seems to function like the type theoretical analogon of the growing domains in model theory: without the rule there is no relation between the inhabitants of set-types in the main and subordinate contexts, and neither Barcan nor Converse Barcan is derivable. This corresponds to Kripke's semantics[3] for

[3] See [Hughes and Cresswell 1972].

modal predicate logic, when no relation is presupposed between the domains of the worlds, and neither formula is valid, regardless of the modal strength of the logic. Adding *transfer*$_3$ makes Converse Barcan and $\exists x \in D.\Box\varphi(x) \supset \Box\exists x \in D.\varphi$ a theorem, just like adopting 'growing domains' in model theory.

However, the discussion of the Barcan formula shows that the analogy between model theory and type theory is not that simple. Given growing domains, Barcan becomes valid in models as soon as the accessibility relation is made symmetrical. Adding growing domains in both directions of the symmetrical accessibility relation yields a model with constant domains, and this is a degenerate case of a model with 'shrinking domains' on which Barcan holds. However, extending the basic modal predicate logic K with the rule for B (or all the way to $KT45$ for that matter) does not make Barcan a theorem of $(\lambda)\Box PRED2$. Also there is no rule in Fitch-style deduction or MPTSs that is an obvious analogon for 'shrinking domains' in Fitch-style deduction or MPTSs, that could be added to make Barcan a theorem.

Given the *transfer*$_3$-rule, and Leibniz identity, Necessity of Identity can be proved in $\lambda\Box PRED2$ analogous to the model theoretic case. This analogy continues to hold for Necessity of Non-Identity, which only becomes a theorem in (extensions of) KB: model theoretically we can only conclude that $a \neq b$ implies $\Box(a \neq b)$ when the domains are kept constant (or shrink). By taking recourse to a weaker form of Leibniz Identity, which precludes 'substitution in the scope of $\Box$', we can block the proof of Necessity of Identity and settle for a Contingent Identity system, where only self-identity is necessary.

The main difference between the model theoretic and type theoretic approach to identity lies in the fact that type theory uses variables for individuals as opposed to the constants used in model theory. This forces a connection between two issues that are unrelated in model theory: we cannot have a system with Necessity of Identity, or even Necessity of Self-Identity, without the Converse Barcan formula since both depend on the *transfer*$_3$ rule. On the other hand it allows us to have contingent identity without $\Box\exists \in D.\varphi(x) \supset \exists x \in D.\Box\varphi(x)$ being a theorem, as under the 'individual concept interpretation' of constants in model theory ([Hughes and Cresswell 1972]).

It is questionable whether the set-variables in an MPTS can play the role of individual in the same way as the constants in model theory. However, we don't seem to have a choice: from a type theoretical point of view constants are just 'free variables that are never bound'. Adding constants for individuals, like we did for the logical axioms, would imply that all individuals are given beforehand since unlike variables constants cannot be created on the spot. Also (like for the axioms) all constants would be available in all subordinate contexts, unless some sort of 'Existence predicate' is defined in the initial context that controls which constants are available at a subordinate context of a given modal depth.

At that point we run into a limitation of the Fitch-style modal deduction, namely that a modal subordinate proof represents an 'arbitrary accessible world'. This means that the associated domain of individuals must somehow be 'representative' for the domains of all worlds that are one step away along the accessibility relation. There can be no 'branching' into several accessible worlds each with their own different set of individuals, as is possible in models. It is to be expected that this deductive limitation, which is inherited by the type systems, is the major factor in restricting the expressive power of predicate logical MPTSs.

From a type theoretical perspective, *transfer*$_3$ is not an essential rule for MPTSs. In the proofs of the meta theoretical properties, it is only needed in cases involving the rule itself. This is different for *transfer*$_1$ and *transfer*$_2$, these rules cannot be missed. They guarantee that propositions, sets and predicates are 'persistent' throughout the subordinate contexts.

Due to the presence of these rules the following 'higher order' Converse Barcan formulas are theorems of $\lambda\Box PRED2$:

$transfer_1$: $\Box(\forall\alpha \in Prop.\varphi) \supset (\forall\beta \in Prop.\Box\varphi)$

$transfer_2$: $\Box(\forall P \in A \rightarrow Prop.\varphi) \supset (\forall Q \in A \rightarrow Prop.\Box\varphi)$,

In terms of possible world semantics the first formula expresses that all possible 'statements' (propositions regardless of their provability) about a world are also 'statements' in accessible worlds of this world (note that this already holds for propositional logic). The second formula states that all properties (predicates) 'available' in a world are 'available' in worlds accessible to that world.

Together they insure that language is preserved when moving to a next world, in every world we can say at least those things that could be said in a previous world (though different propositions may be provable). In this way the rules that preserve well-typedness in the MPTSs bring an implicit assumption of modal predicate logic into the formalism, namely (the stronger assumption) that the language (the sets of well-formed formulas and terms) is the same in every world. In the alethic interpretation where '$\Box$' is a necessity operator this seems a reasonable enough assumption. However, in other intensional readings of the operator it may be less reasonable, for instance in multi-agent epistemic logic where agents can reason about each others knowledge and beliefs: there something which is a proposition (has type *Prop*) to one agent may not be a proposition to another. In the MPTSs we can begin to discuss these assumptions about the language formally, by investigating the effects of manipulating the transfer of well-typedness information.

Chapter 6

Contexts in dialogue

According to the preface of this thesis, the MPTSs presented in the previous chapters were developed with type theoretical knowledge representation in mind. This chapter is intended to give the reader an impression of how MPTSs fit in with existing ideas on the formalization of communication. For lack of a full-grown type theoretical account of communication, we devise our own formalization of one small aspect: the 'update' of the information state of a hearer-agent by a declarative utterance of a speaker-agent. This procedure is based on the MPTS $\lambda\Box PRED2$ (section 6.1,6.2), and meant merely as a finger exercise in dialogue formalization. Although the procedure is a little naïve, it does show that existing work on pragmatics in epistemic/doxastic logic ([Thijsse 1992], section 6.4) and on the representation of natural language ([Kamp 1981], [Ahn and Kolb 1990], section 6.3) can be brought together in the MPTS-framework (section 6.5). Besides, it will serve as a guideline for further discussion (section 6.6).

6.1. Contexts as growing information states

In [Ahn 1992] a type theoretical approach to user modelling in man-machine communication is proposed. Central to this proposal is the idea that the information state of an agent (animate or inanimate) can be modelled by a type theoretical context. In this view, the assertions that make up an agent's information state are represented as statements, where the type of a statement corresponds to an assertion of the agent and the term inhabiting the type corresponds to the 'justification' or 'evidence' the agent has for this assertion. In general, the information state of an agent will not contain a complete (or even accurate) description of the world: an agent may be uncertain about some propositions and unaware of others. Since the information state is incomplete, it may 'grow' as the agent learns more about the world. This growth can be modelled by appending statements representing the new information to the context representing the agent's information state.

One source of growth is communication between agents, and Ahn ([Ahn 1992]) sketches a perspective under which dialogue can be viewed type theoretically as an exchange of information between (growing) contexts. It is assumed that the participants in the dialogue exchange information only through utterances, like in a telephone conversation. Against the background of Ahn's ideas, we construct a procedure for a particular instance of information growth in dialogue: the 'update' of the information state of a hearer-agent by a declarative utterance of a speaker-agent.

We formulate this update in a simple dialogue situation involving two agents, a speaker (S) and a hearer (H). Since we consider the effect of a single utterance of the speaker, the agents have fixed roles: the speaker speaks, and the hearer listens. The information states of the speaker and the hearer are represented as (non-blocked) contexts in $\lambda\Box PRED2$. Such a context contains declarations of all entities that the agent assumes to exist, and of all assertions (along with their proofs) that he holds about the world.

It also contains statements declaring the 'vocabulary' (predicates, functions, sets) in which these assertions are formulated. Assuming that these statements denote concepts that are somehow related to words in the language, agents speaking the same language must share a considerable amount of this vocabulary to make communication possible. Ahn envisions a formalization in which this shared vocabulary is represented by a *common context* which can be extended during dialogue, and of which each participant maintains his own version. Ideally, these two versions are isomorphic. When large discrepancies exist between them, misunderstandings will arise.

We want to abstract from such misunderstandings, and hence simply make the participants' information states before the dialogue (their *initial contexts*) isomorphic, by assuming that they contain the same vocabulary. To make this assumption more precise, we recall the partition of the terms of (M)PTSs in the (Modal) Logic Cube defined in section 3.3:

6.1.1. DEFINITION. **Partition of pseudoterms**

The terms of the *PTS*s in the Logic Cube can be partitioned into:

i A is a *set-kind*: $\exists\Gamma[\Gamma \vdash A : \mathit{Type}^s]$

ii A is a *prop-kind*: $\exists\Gamma[\Gamma \vdash A : \mathit{Type}^p]$

iii A is a *set-constructor*: $\exists\Gamma, B[\Gamma \vdash A : B : \mathit{Type}^s]$

iv A is a *prop-constructor*: $\exists\Gamma, B[\Gamma \vdash A : B : \mathit{Type}^p]$

v A is a *set*: $\exists\Gamma[\Gamma \vdash A : \mathit{Set}]$

vi A is a *proposition*: $\exists\Gamma[\Gamma \vdash A : \mathit{Prop}]$

vii A is an *element*: $\exists\Gamma, B[\Gamma \vdash A : B : \mathit{Set}]$

viii A is a *proof term*: $\exists\Gamma, B[\Gamma \vdash A : B : \mathit{Prop}]$

The terms making up the vocabulary, predicates, functions, sets, etc. are all in categories $i - vi$, which allows us to express the assumption that the dialogue participants have the same vocabulary as follows:

The Vocabulary Assumption All agents in *People* have the same *kinds*, *constructors*, *sets* and *propositions* declared in their initial context.

Under this assumption, the initial context of agents can only differ in the *elements* and *proofs*. This means that speaker and hearer may be able to prove different propositions, and may have different proofs for the same proposition. Similarly, they may be familiar with different elements of a given set, or differ in the sets that are inhabited for them.

In MPTSs, the only way to add a statement to a context is by an application of the *start* or *weakening*-rule (cf. chapter 3.1). Hence any procedure for adding statements to a context 'from the outside' should result in a context which looks as if it *could* have been derived from the original context by applications of these two derivation rules. Looking at *start* and *weakening*, we can see that this amounts to the following requirements for adding a statement $x : A$, representing new information, to a context Γ, representing the information state of an agent:

(i) $\Gamma \vdash A : s$, A is well-typed on Γ,

(ii) x is 'Γ-fresh', it does not yet occur in Γ.

Given the discussion above, it will be clear that the well-typedness requirement (i) ensures that the agent can 'understand' the new information. The second requirement can be seen as the type theoretical reflection of a fundamental impossibility in communication: by an utterance, a speaker communicates a certain (propositional) content, but he cannot by the same utterance convey his evidence for this content to the hearer. Hence, if we equate the content of an utterance with the type of a statement, the proof object is 'lost' in communication. The agent that adds the new information to his context replaces the lost proof object with a 'dummy' proof object, a fresh variable that merely signifies that the type is inhabited.

The types of the statements that we will be adding to the context of the hearer in the update operation correspond to terms in the logic $\Box PRED2$. Using the definition of term-context (cf. section 2.3) for this logic, the context needed to prove that such a term is well-typed can be constructed inductively.

6.1.2. DEFINITION. **Term-contexts**
For every term t of the language of $\Box PRED2$ we define a context Γ_t such that $\Gamma_t \vdash t : D$ (in $\lambda\Box PRED2$) if $t \in D$ (in $\Box PRED2$), as follows

$$\begin{aligned}
t \equiv x^D &\Rightarrow \Gamma_t := x^D : D,\\
t \equiv \lambda x \in D.M &\Rightarrow \Gamma_t := \Gamma_M/(x : D),\\
t \equiv MN &\Rightarrow \Gamma_t := \Gamma_M \cup \Gamma_N,\\
t \equiv \varphi \supset \psi &\Rightarrow \Gamma_t := \Gamma_\varphi \cup \Gamma_\psi,\\
t \equiv \forall x \in D.\varphi &\Rightarrow \Gamma_t := \Gamma_\varphi/(x : D),\\
t \equiv \Box\varphi &\Rightarrow \Gamma_t := \Gamma_\varphi.
\end{aligned}$$

Given this definition, requirement (i) can be rephrased as follows: a statement $x : A$ can be added to a $\lambda\Box PRED2$-context Γ if $\Gamma_A \subseteq \Gamma$. In that case $\Gamma \vdash A : s$, and hence the extended context $\Gamma, x : A$ could have been obtained by an application of *start* (provided x is Γ-fresh).

Note that the Vocabulary Assumption does not guarantee that $\Gamma \vdash A : s$ for arbitrary A on any Γ representing the information state of an agent in *People*, since A (and hence Γ_A) may contain occurrences of *elements* or *proofs* that are not declared in Γ.

6.2. Multi-agent modal predicate logic

In chapter 4, it was shown how reasoning of agents about the knowledge and beliefs of other agents can be accommodated in MPTSs by decorating the subordinate contexts, modalities and modal rules with an agent index. By indexing the MPTS $\lambda\Box PRED2$ in this way, a system for multi-agent modal predicate logic is obtained. In this multi-agent system, the interaction between quantifiers and modal operators can be quite complex; not only do we have to deal with quantification over a domain in contexts of different modal depth, these contexts may also have different agent indices.

The upshot of the discussion in chapter 5 of the interaction between quantification and modal operators is that, for mono-logical predicate logic, the crucial choice is whether to adopt the rule *transfer*$_3$. This rule guarantees that elements of sets ('objects', 'individuals') remain available when moving to a subordinate context. If *transfer*$_3$ is adopted, the Converse Barcan formula becomes a theorem of $\lambda\Box PRED2$. This is a reasonable principle for a system in which one agent (say agent 2) reasons about his own knowledge or beliefs; given a proof z of $\Box_2\forall x \in D.\varphi(x)$, agent 2 can derive a proof $(\lambda y : A.(\hat{k}_2((\check{k}_2 z)y)))$ of $\forall x \in D.\Box_2\varphi(x)$ since all 'D-objects' that he is familiar with remain available in the subordinate context. In a multi-agent system another agent, say 1, can make the same inference if his information state contains a proof that 2 knows or believes $\forall x \in D.\varphi(x)$. However, the inference made by 1 turns a proposition which is about the 'D-objects' that 2 is familiar with ($\Box_2\forall x \in D.\varphi(x)$) into a proposition ($\forall x \in D.\Box_2\varphi(x)$) which quantifies over 'D-objects' that 1 is familiar with, since the quantifier now has the greater scope and the statement is derived on 1's context. Hence in this multi-agent case, Converse Barcan expresses a relation between the inventory of the context of 1 and the inventory of the (subordinate) context(s) of 2. So far we have not presumed anything about the relation between the various domains of the different agents. The fact that *transfer*$_3$ induces such a relation is an argument for not including it in the multi-agent version of $\lambda\Box PRED2$ but, as will become clear from the following examples, there are also reasons for adopting it.

The unproblematic cases are those where an agent, reasoning about the knowledge or beliefs of another agent, derives only universally quantified conclusions in the subordinate context. In the following example, the context of agent 1 contains evidence that 2 believes that $\forall x \in D.(Px \supset Qx)$ and $\forall x \in D.Px$. He can infer that 2 believes that $\forall x \in D.Qx$ in the following way:

1. $\Gamma, z : \Box_2(\forall x : D.Px \supset Qx), u : \Box_2\forall x : D.Px \vdash z : \Box_2\forall x : D.Px \supset Qx$ $\quad (\Gamma \vdash D : Set)$
2. $\Gamma, z : \Box_2(\forall x : D.Px \supset Qx), u : \Box_2\forall x : D.Px \vdash u : \Box_2\forall x : D.Px$
3. $\Gamma, z : \Box_2(\forall x : D.Px \supset Qx), u : \Box_2\forall x : D.Px \boxdot_2 \varepsilon \vdash \check{k}_2 z : \forall x : D.Px \supset Qx$
4. $\Gamma, z : \Box_2(\forall x : D.Px \supset Qx), u : \Box_2\forall x : D.Px \boxdot_2 \varepsilon \vdash \check{k}_2 u : \forall x : D.Px$
5. $\Gamma, z : \Box_2(\forall x : D.Px \supset Qx), u : \Box_2\forall x : D.Px \boxdot_2 \varepsilon \vdash D : Set$ $\quad$ (transfer$_1$)
6. $\Gamma, z : \Box_2(\forall x : D.Px \supset Qx), u : \Box_2\forall x : D.Px \boxdot_2 y : D \vdash y : D$
7. $\Gamma, z : \Box_2(\forall x : D.Px \supset Qx), u : \Box_2\forall x : D.Px \boxdot_2 y : D \vdash (\check{k}_2 z)y : Py \supset Qy$
8. $\Gamma, z : \Box_2(\forall x : D.Px \supset Qx), u : \Box_2\forall x : D.Px \boxdot_2 y : D \vdash (\check{k}_2 u)y : Py$
9. $\Gamma, z : \Box_2(\forall x : D.Px \supset Qx), u : \Box_2\forall x : D.Px \boxdot_2 y : D \vdash ((\check{k}_2 z)y)((\check{k}_2 u)y) : Qy$
10. $\Gamma, z : \Box_2(\forall x : D.Px \supset Qx), u : \Box_2\forall x : D.Px \boxdot_2 \varepsilon \vdash \lambda y : D.((\check{k}_2 z)y)((\check{k}_2 u)y) : \forall y : D.Qy$
11. $\Gamma, z : \Box_2(\forall x : D.Px \supset Qx), u : \Box_2\forall x : D.Px \vdash \hat{k}_2(\lambda y : D.((\check{k}_2 z)y)((\check{k}_2 u)y)) : \Box_2\forall y : D.Qy$

First agent 1 K-imports the $\Box_2$-statement into a 2-subordinate context where their types appear as universal formulas quantifying over 2's domain D (lines 3,4). Then 1 assumes an arbitrary element (y) of type D (like 2 could have done reasoning about his own knowledge), to instantiate the universal formulas and derive a proof of Qy (lines 6-9). Then object assumption is discharged and the conclusion $\forall y \in D.Qy$ (quantifying over 2's domain) is brought back to its context in the form $\Box_2 \forall y \in D.Qy$. In cases like these, *transfer*$_3$ does not come into play, and there is no 'confusion of domains'.

Of course not all reasoning in subordinate contexts leads to universal conclusions. For instance given a context of agent 1 which contains proofs for '2 believes that $\forall x : D.Px \supset Qx$' and '2 believes that Pa'. In this case it seems reasonable for 1 to infer that '2 believes that Qa', since the fact that 2 believes that Pa indicates that 2 is familiar with the object a. However, even with $a : D$ present in 1's context, $\Box_2 Qa$ cannot be proved.

1. $\Gamma, z : \Box_2(\forall x : D.Px \supset Qx), u : \Box_2 Pa, a : D \vdash z : \Box_2 \forall x : D.Px \supset Qx$
2. $\Gamma, z : \Box_2(\forall x : D.Px \supset Qx), u : \Box_2 Pa, a : D \vdash u : \Box_2 Pa$
3. $\Gamma, z : \Box_2(\forall x : D.Px \supset Qx), u : \Box_2 Pa, a : D \vdash a : D$
4. $\Gamma, z : \Box_2(\forall x : D.Px \supset Qx), u : \Box_2 Pa, a : D \boxdot_2 \varepsilon \vdash \check{k}_2 z : \forall x : D.Px \supset Qx$
5. $\Gamma, z : \Box_2(\forall x : D.Px \supset Qx), u : \Box_2 Pa, a : D \boxdot_2 \varepsilon \vdash \check{k}_2 u : Pa$

The problem is that without *transfer*$_3$ there is no way to instantiate $\forall x : D.Px \supset Qx$ in the 2-subordinate context with the $a : D$ from 1's context. With *transfer*$_3$, deriving a proof of $\Box_2 Qa$ is straightforward.

6. $\Gamma, z : \Box_2(\forall x : D.Px \supset Qx), u : \Box_2 Pa, a : D \boxdot_2 \varepsilon \vdash a : D$ (transfer$_3$ 3)
7. $\Gamma, z : \Box_2(\forall x : D.Px \supset Qx), u : \Box_2 Pa, a : D \boxdot_2 \varepsilon \vdash (\check{k}_2 z)a : Pa \supset Qa$
8. $\Gamma, z : \Box_2(\forall x : D.Px \supset Qx), u : \Box_2 Pa, a : D \boxdot_2 \varepsilon \vdash ((\check{k}_2 z)a)(\check{k}_2 u) : Qa$
9. $\Gamma, z : \Box_2(\forall x : D.Px \supset Qx), u : \Box_2 Pa, a : D \boxdot_2 \varepsilon \vdash ((\check{k}_2 z)a)(\check{k}_2 u) : Qa$
10. $\Gamma, z : \Box_2(\forall x : D.Px \supset Qx), u : \Box_2 Pa, a : D \vdash \hat{k}_2(((\check{k}_2 z)a)(\check{k}_2 u)) : \Box_2 Qa$

As remarked above, *transfer*$_3$ induces an inclusion relation between the inventory of the 2-subordinate context and the context of 1: every element available in 1's context is available in the 2-subordinate context. Sometimes this is desirable (like in the previous example), sometimes it is not. The following example depicts a situation which brings out this ambiguity. The context of agent 1 contains an object $a : D$, as well as evidence that agent 2 believes that $\forall x : D.Px$. By means of *transfer*$_3$, 1 can use $a : D$ to instantiate $\forall x : D.Px$ in the 2-subordinate context and hence obtain a proof that 2 believes that Pa.

1. $\Gamma, u : \Box_2 \forall x : D.Px, a : D \vdash u : \Box_2 \forall x \in D.Px$
2. $\Gamma, u : \Box_2 \forall x : D.Px, a : D \vdash a : D$
3. $\Gamma, u : \Box_2 \forall x : D.Px, a : D \boxdot_2 \varepsilon \vdash \check{k}_2 u : \forall x : D.Px$
4. $\Gamma, u : \Box_2 \forall x : D.Px, a : D \boxdot_2 \varepsilon \vdash a : D$
5. $\Gamma, u : \Box_2 \forall x : D.Px, a : D \boxdot_2 \varepsilon \vdash (\check{k}_2 u)a : Pa$
6. $\Gamma, u : \Box_2 \forall x : D.Px, a : D \vdash \hat{k}_2((\check{k}_2 u)a) : \Box_2 Pa$

Whether this inference is intuitively correct or not depends on the familiarity of 2 with the individual a. If a is an element of 2's (subordinate context-) domain, the inference is correct. If it is not ($a : D$ is just familiar to agent 1), 1 should not be able to derive that 2 believes something about an object that he is not familiar with (that is not in 2's domain). In other words, it should only be allowed to bring an object to the 2-subordinate context when this object is an element of 2's domain.

Refining *transfer*$_3$ in this way requires that objects in 1's context that are familiar to 2 are syntactically discernable from objects that are not. Hence we propose to index the elements $(x : D : Set)$ occurring in the context of an agent with respect to the (other) agents that are familiar with these elements. In cases like the above, where we have two agents, this means that every element in 1's context may be labelled with the agent index 2, indicating that 2 is familiar with this object (according to 1). Once the individuals are indexed with respect to agents, we can refine *transfer*$_3$ in the desired way by requiring that the term (A) of the statement $(A : B(: Set))$ that is to be transferred has the same agent index as the subordinate context in which it is to be reiterated.

$$transfer_3^a \frac{G \vdash A_a : B : Set}{G \boxdot_a \varepsilon \vdash A_a : B}$$

Under this rule, the relation between the inventory of a context and its subordinate contexts expressed by Converse Barcan only holds for elements that have the same agent index as the subordinate context. Hence Converse Barcan is no longer a theorem, but it still holds 'individually' for any agent reasoning about his own knowledge or beliefs.

Of course a lot more would have to be specified about the calculus of agent-indices on statements (like how they are created etc.) in relation to $transfer_3^a$, but for the moment we will use the rule as specified above in settings where the labelling is given, and where we assume that the labels do not interfere with any other rule.

Confusion of domains can also be forced by modal means, i.e. the T-export rule. If we go back to the first example in this section, and interpret '$\Box_2$' as '2 knows', then 1 could have applied T-export instead of K-export, resulting in the following last line of the derivation:

11. $\Gamma, z : \Box_2(\forall x : D.Px \supset Qx), u : \Box_2 \forall x : D.Px \vdash \hat{k}_2(\lambda y : D.((\check{k}_2 z)y)((\check{k}_2 u)y)) : \forall y : D.Qy$

Here agent 1 has reached a conclusion about his own domain $(\forall y : D.Qy)$ by reasoning about 2's knowledge. It is easy to check that the conclusion does not depend on the predicate logical reasoning of 1 in the 2-subordinate context. Agent 1 could have reached the same conclusion by using only 'propositional steps' with respect to 2's knowledge. By first assuming $y : D$ in his own context and then subsequently applying K-import and T-export to both $\Box_2$-formulas, 1 can obtain the proof $\lambda y : D.(((\hat{t}\check{k}z))y)((\hat{t}(\check{k}u))y)$ of $\forall y : D.Qy$[1]. Agent 1's conclusion is only valid if the $D - elements$ familiar to him are a subset of the D-elements agent 2 is familiar with. Hence the T-export rule presupposes a relation between the domains of the agents, independent of *transfer*$_3$.

Since the T-axiom is commonly held to express the property that separates knowledge from belief, the above observations points in the direction of an asymmetry between predicate logical belief and predicate logical knowledge: multi-agent doxastic predicate logic is compatible with contingent relations between the domains of the agents, multi-agent epistemic predicate logic is not.

6.3. Discourse Representation Theory in type theory

The Discourse Representation Theory (DRT) of Hans Kamp ([Kamp 1981]) is a formal method for constructing representations for texts (sequences of sentences) in three steps. Starting from the sentences in the discourse a 'Discourse Representation Structure' (DRS) is generated,

[1] The second term proving $\forall y : D.Q(y)$ can be obtained from the first by means of $\hat{t}$-reduction(cf. chap 2.6).

processing them 'from left to right' by means of 'DRS-construction rules'. These structures are then interpreted in a model through a truthful embedding. In this section we shall not go into the construction nor into the embedding of DRSs, since our main concern is the relation of (already constructed) DRSs to type theory. In [Ahn and Kolb 1990] a formal translation is given from DRSs into type theoretical contexts: each DRS corresponds to a 'segment' (a small pseudo-context). Using this translation, the growth of the information state of an agent interpreting a text can be modelled by the extension of the context representing the agent's information state with the segment representing the text.

DRT is focussed on the resolution of anaphoric ambiguities, both within sentences and across sentence boundaries. To do this, formal individuals called 'discourse referents' are introduced which may serve as antecedents for pronouns. For example, suppose we want to represent the two-sentence discourse 'A farmer owns a donkey. He beats it'. After processing the first sentence, the discourse representation looks like this[2].

$u \quad v$
$farmer(u)$
$donkey(v)$
$owns(u, v)$

This representation contains the two formal individuals (u and v) introduced by the sentence, along with the information that the first individual is a farmer, the second a donkey, and the fact that the first individual owns the second. In the second sentence of the discourse ('He beats it'), both 'he' and 'it' refer back to the first sentence. To incorporate the information contained in the second sentence, we first have to decide what these references are. In view of the fact that the entire preceding discourse consists of the previous sentence, the natural reading is that in which 'he' refers to the farmer, and 'it' to the donkey. Hence we extend the above representation with an entry stating that u beats v.

$u \quad v$
$farmer(u)$
$donkey(v)$
$owns(u, v)$
$beats(u, v)$

Kamp calls figures of this simple kind 'Discourse Structures'(DRs). Besides discourse referents and 'simple conditions' (predications over referents), DRs may also contain 'links', which equate a discourse referent with a name (or another discourse referent). We change the example discourse above to one that involves names to illustrate the use of links. The discourse 'Pedro is a farmer. Jerry is a donkey. Pedro owns Jerry and beats him' is represented as:

[2]We use the 'stripped' format of [Van Eijck 1985], rather than the original format of [Kamp 1981].

$u \quad v$
$u = Pedro$
$farmer(u)$
$v = Jerry$
$donkey(v)$
$owns(u, v)$
$beats(u, v)$

As we indicated earlier, DRT is also used for the resolution of anaphora inside sentences. As a matter of fact, this was one of its strong points, since it could handle anaphoric dependencies that were beyond the scope of theories existing at that time. The most famous example of this is the so-called 'donkey-sentence': 'Every farmer who owns a donkey beats it'. The problem with this sentence is that 'it' refers to different donkeys, depending on the respective farmer who has ownership and performs the beating. The construction rule for universally quantified sentences assigns the donkey-sentence a representation that consists of three DRs.

$u \quad v$ $farmer(u)$ $donkey(v)$ $owns(u, v)$	$u \quad v$ $beats(u, v)$

Kamp refers to compound representations like this as 'Discourse Representation Structures'. The upper (empty) DR is the *principal* DR of the DRS, the DRs below are *subordinate* to the upper one. The lower right DR is in turn subordinate to the lower left DR. The idea expressed by these structural dependencies is that a DRS of this form can be truthfully embedded in a model iff every truthful embedding in the model of the lower left DR can be extended to a truthful embedding of the right DR. In other words, for every pair of entities in the model consisting of a farmer and a donkey where the farmer owns the donkey, the model validates that the farmer beats the donkey.

Ahn and Kolb do not give a direct translation of the two-dimensional representations into type theoretical contexts. They use an intermediate sequential format in which *DRS*s are written in the following form: $r_1, \ldots, r_n, E_1, \ldots, E_m$, where $(r_1, \ldots, r_n)$ are the discourse referents and the 'entries' $E_1, \ldots, E_m$ are of one of the following three forms:

- atomic condition, n-ary predicate applied to a number of discourse referents,
- a complex condition $D_1 \Rightarrow D_2$, where D_1 and D_2 are DRSs,
- a link $[R = N]$ or $[R = R']$, where R and R' are discourse referents, and N is a name in the model.

It will be clear that *DRs*, which consist only of a box with discourse referents, simple conditions, and links, can be represented in this format. However, the case for complex conditions deserves some explanation. For instance, for the donkey sentence we obtain an entry $D_1 \Rightarrow D_2$ where D_1 and D_2 correspond to *DRs*. These separate components are represented sequentially as

$D_1 : u, v, farmer(u), donkey(v), owns(u, v)$
$D_2 : u, v, farmer(u), donkey(v), owns(u, v), beats(u, v)$.

To represent the complex condition we only have to look at the entries, since D_1 and D_2 have the same discourse referents. The dependency between the entries is that if $farmer(u)$, $donkey(v)$, and $owns(u, v)$, then $beats(u, v)$. Hence we can represent $D_1 \Rightarrow D_2$ by means of the following sequence:

$$u, v, (farmer(u), donkey(v), owns(u, v) \Rightarrow beats(u, v)).$$

Since this covers all available constructions, we can use the sequential format to represent *DRSs* in the form $r_1, \ldots, r_n, E_1, \ldots, E_m$.

Given such a sequential representation of a *DRS*, Ahn and Kolb propose the following translation of DRSs to type theoretical contexts: a sequence of the general form $r_1, \ldots, r_n$, $E_1, \ldots, E_m$ translates to a 'segment' of the general form $r1 : entity, \ldots, r_n : entity$, $y_1 : E_1, \ldots, y_m : E_m$. The discourse referents are translated directly into variables. This is in line with the intuition that set variables act as 'pointers', they make an object of a certain type available to the reasoner. Since *DRT* has no typing (properties are attributed to the referents via predication), we give all discourse referents the same (neutral) type 'entity'.

Entries are translated as terms of type *Prop* and they get a fresh variable $(y_1, \ldots, y_m)$ assigned as their proof term; the entries represent the content of the discourse, not its justification. The three kinds of entries are accommodated in $\lambda\Box PRED2$ as follows. Atomic conditions are an n-ary predicate applied to a number of referents. As in chapter 5, these are translated to $P(r_1, \ldots, r_n) : Prop$. Complex conditions are of the form $D_1 \Rightarrow D_2$. Roughly speaking, they are translated as a (series of) Π-abstraction(s) connecting D_1 to (part of) D_2. We illustrate this by means of the donkey sentence. For this sentence, the segments corresponding to D_1 and D_2 are

$D_1 : \ u : entity, v : entity, p_1 : farmer(u), p_2 : donkey(v), p_3 : owns(u, v)$
$D_2 : \ u : entity, v : entity, p_1 : farmer(u), p_2 : donkey(v), p_3 : owns(u, v), p_4 : beats(u, v)$.

The sequence $u, v, (farmer(u), donkey(v), owns(u, v) \Rightarrow beats(u, v))$ $(D_1 \Rightarrow D_2)$ is translated into the statement $z : (\Pi u : entity.\Pi v : entity.\Pi p_1 : farmer(u).\Pi p_2 : donkey(v).\Pi p_3 : owns(u, v).beats(u, v))$, where the Πs abstract over the elements of the segment corresponding to D_1, and the body of the abstraction is the D_2-segment minus the statements that are also in the D_1-segment (and z is a fresh variable). This abstraction is the proof theoretical reflection of the semantical idea that $D_1 \Rightarrow D_2$ turns any assignment satisfying D_1 into an assignment satisfying D_2. An interpreter who already has entities (x, y) in his context as well as proof that these entities are respectively a farmer and a donkey $(p_5 : farmer(x), p_6 : donkey(y))$ and that x owns y $(p_7 : owns(x, y))$, can derive a term $z(x, y, p_5, p_6, p_7)$ proving 'x beats y' $(beats(x, y))$ by applying all this information to the type theoretical translation of $D_1 \Rightarrow D_2$. Links are expressions of the form $R = R'$ or $R = N$, which 'link' a discourse referent R to another discourse referent (R') or a name in the model (N). They can be expressed type theoretically by means of Leibniz Identity; $y : (R = R')(: Prop)$ or $y : (R = N)(: Prop)$. The difficulty with links of the latter kind is to find the type theoretical equivalent of proper names in the model. Ahn and Kolb simply use (set-) variables; *Pedro* : *entity*, *Jerry* : *entity* Although it is unclear whether this is satisfactory in the setting of modal reasoning, we represent the names Pedro and Jerry in this way in the following examples assuming that these 'name-variables' are somehow given beforehand (they are declared in the initial context of every agent).

If adding a segment $r_1 : entity, \ldots, r_n : entity, y_1 : E_1, \ldots, y_m : E_m$ to the context of the interpreter is to result in a legal pseudo context, the following well-typedness constraints have to be met (assuming that all variables occurring as subjects in the segment are fresh).

- The variables $r_1, \ldots, r_n$, representing the discourse referents, are all of type *entity*. Hence all that is required for their well-typedness is that this type is declared in the context of the interpreter: $\Gamma \vdash entity : Set$. Hence we have $\Gamma, r_1 : entity, \ldots, r_i : entity \vdash entity : Set$ for all $i : 1 \leq i \leq n$.

- The well-typedness of the entries has to be settled individually. For the first entry, it has to be the case that $\Gamma, r_1 : entity, \ldots, r_n : entity \vdash E_1 : Prop$. Each of the following entries may then depend on its predecessors, giving rise to the following general well-typedness requirement: $\Gamma, r_1 : entity, \ldots, r_n : entity, y_1 : E_1, \ldots, y_j : E_j \vdash E_{j+1} : Prop$ (for $j : 1 \leq j < m$).

If these conditions are met, the segment is called a *valid extension* of the context of the interpreter. The well-typedness information already present in this context is sufficient to account for the type-variables that occur freely in the entries. Intuitively the above requirements express that the interpreter possesses sufficient linguistic means to 'understand' the discourse content represented in the segment. If the second condition is not met for one of the entries ($\Gamma, r_1 : entity, \ldots, r_n : entity, y_1 : E_1, \ldots, y_j : E_j \not\vdash E_{j+1} : Prop$), the discourse contains a 'concept' (predicate, noun) that is not familiar to the interpreter.

The 'anchoring' of the discourse representation in the information state of the interpreter can be taken a step further by letting the interpreter replace the 'fresh' referents and proof variables in the segment with objects already present in (or derivable on) his context. This idea can be taken up in various ways. A grand but illustrative perspective is that of Ranta who is particularly interested in the representation of literary texts. In [Ranta 1989], anchoring is seen as the process that can account for the fact that every reader of literary work has his own interpretation, even though there exists a (DRT-like) canonical type theoretical representation. The difference between the interpretations lies in the way each reader ankers characters, locations etc. occurring in the text by substituting them with persons and places from his own mental state. On a more technical level, [Ahn and Kolb 1990] discusses anchoring of a segment in the context of the interpreter as the proof theoretical analogon of the embedding of a *DRS* in a model.

6.4. Epistemic pragmatics

Reasoning about information states of (other) agents plays an important role in communication. For instance, in an information dialogue it is not cooperative to ask your dialogue partner something you already know, or to ask him a question you know he cannot answer. A famous attempt to codify 'cooperative' behaviour in dialogue was made by Grice (see for instance [Grice 1989]). He begins his top-down development of dialogue behaviour rules by stating the

> **Cooperation principle** Make your conversational contribution such as is required, at the stage at which it occurs, by the accepted purpose or direction of the talk exchange in which you are engaged.

Starting from this principle, Grice discerns four categories of rules for dialogue behaviour ('maxims'), each characterized by a 'super maxim':

> **Quantity** Make your contribution as informative as is required
>
> **Quality** Try to make your contribution one that is true
>
> **Relevance** Be relevant
>
> **Manner** Be perspicuous

Although no order is imposed on these categories, Grice does remark that the maxims of quality are more important than maxims of the other categories, since the latter can only come into play after the maxims of quality have been satisfied. The general advice to 'try to make your contribution one that is true' is then specified further in two maxims:

> **Belief** Do not say what you believe to be false.
>
> **Evidence** Do not say something for which you lack sufficient evidence.

In [Thijsse 1992], the 'epistemic force' that is attributed to (declarative) utterances through the quality maxims is analyzed in terms of epistemic/doxastic logic. This analysis results in the following proposal for an 'utterance rule'.

$$UTT \quad x : `\varphi' \ \Rightarrow B_x K_x \varphi.$$

If an agent (x) utters the proposition φ (x :'φ') he should believe to know that φ, φ should be a true justified belief of his. An important benchmark in the epistemic analysis of the quality maxims are Moore's paradoxes (cf. [Moore 1912]), sentences about self-belief of the kind

(1) p, but I do not believe that p: $p \wedge \neg B_i p$

(2) p, but I believe that not p: $p \wedge B_i \neg p$

The puzzling thing about these sentences is that although they are logically consistent (the logical translations given above have verifying models), they are absurd to utter. In [Hintikka 1962] a similar example involving self-knowledge is given

(3) p, but I do not know whether p: $p \wedge \neg K_i p \wedge \neg K_i \neg p$.

The peculiarity of these 'Moore-sentences' can be formally demonstrated after the application of UTT to their logical translations: the resulting formulas are inconsistent in epistemic/doxastic logic.

The epistemic/doxastic system used by Thijsse is a multi-modal multi-agent system with logic $KT4_{(m)}$ for knowledge and $KD4_{(m)}$ for belief, where knowledge and belief are linked by the axiom $K_a\varphi \supset B_a\varphi$ (corresponding to the rule *FK*-import). In this system each of the above formulas can be proved to be inconsistent under *UTT*. As an example we show this for 'φ but I don't believe that φ' (representing the deictic 'I' by the agent index i):

$$(\varphi \wedge \neg B_i\varphi) \Rightarrow_{UTT} B_iK_i(\varphi \wedge \neg B_i\varphi).$$

1.	$B_iK_i(\varphi \wedge \neg B_i\varphi)$	
	$\boxed{B_i}$	
2.	$K_i(\varphi \wedge \neg B_i\varphi)$	(K-import 1)
	$\boxed{K_i}$	
3.	$\varphi \wedge \neg B_i\varphi$	(K-import 2)
4.	φ	
5.	$\neg B_i\varphi$	
6.	$K_i\varphi$	(K-export 4)
7.	$K_i\neg B_i\varphi$	(K-export 5)
	$\boxed{K_i}$	
8.	$\neg B_i\varphi$	(K-import 7)
9.	$K_i\varphi$	(4-import 6)
	$\boxed{B_i}$	
10.	φ	(FK-import 9)
11.	$B_i\varphi$	(K-export 10)
12.	$\bot$	
13.	$\bot$	(D'-export 12)
14.	$\bot$	(D'-export 13)

(To shorten this derivation we use a derived rule, D'-export[3], in the last two lines). The example shows how a speaker can derive that uttering a Moore-sentence is inconsistent, given *UTT* and the rules of the epistemic/doxastic logic.

Moore-sentences are not only strange to utter, they are also strange to hear. The analysis of the epistemic force of utterances should account for this in terms of the effects of an utterance on the information state of the hearer. In general a hearer need not be convinced of what the speaker says, but it seems reasonable to assume that the hearer is convinced that the speaker is convinced of what he says. Thijsse calls this effect 'epistemic transfer', and he extends his proposal accordingly with the following rule describing this effect of uttering a proposition (φ) by the speaker (x) on the hearer (y):

$$\textit{epistemic transfer}\ x : `\varphi' \;\Rightarrow\; B_yK_yB_xK_x\varphi$$

The combination of modal operators in front of φ shows that the hearer is as sure of the utterance of the speaker, $B_HK_H(B_SK_S\varphi)$, as the speaker is of the proposition he utters, $B_SK_S(\varphi)$. In other words, the rule *UTT* is available to the hearer and is internalized by him.

[3] This rule allows falsehood $\bot$ (and only this formula) to be brought unchanged from the subordinate to the main proof. Deriving it in *KD* requires the use of '$\top$' ($=_{def} \bot \supset \bot$), the 'always true' formula, which may be written anywhere in a proof without further justification.

With respect to the Moore-sentences, the *epistemic transfer*-rule does for the hearer what *UTT* does for the speaker, it lets the hearer derive that the utterance of these sentences by the speaker is inconsistent in the epistemic/doxastic logic (along with some weaker conclusions such as the belief that the speaker believes a falsehood ($B_H B_S \bot$)). More generally, it can be shown that the propositions that lead to inconsistency under *UTT* are inconsistent under *epistemic transfer* and vice versa[4]:

$B_S K_S \varphi$ is consistent iff $B_H K_H B_S K_S \varphi$ is consistent.

Although there is a lot more to be said about the quality maxims (we could for instance go on to analyze the effects of an utterance on the hearer *as perceived by the speaker*), we close the discussion here since we got what we came for; a modal formalization of the effects of an utterance on the information state of the hearer. The epistemic/doxastic logic in which *UTT* and *epistemic transfer* 'live' can be accommodated in an MPTS, it is a subsystem of the multi-modal logic KB_{CD} discussed in chapter 4. Hence the derivations made by speaker and hearer based on these rules have a counterpart in modal type theory. What remains to be done is incorporating the 'modalization' of uttered propositions prescribed by the pragmatic rules in a procedure for adding (type theoretical representations of) utterances to the context of the hearer.

6.5. Adding declarative utterances

In this section, the ingredients presented separately above are combined into a procedure for adding declarative utterances to the information state of the hearer.

Starting from a declarative utterance of the speaker, a type theoretical representation of its content can be obtained by taking what the speaker says (the sentence used) to be a discourse. For this discourse a *DRS* can be constructed, which is turned into a segment, $r_1 : entity, \ldots, r_n : entity, y_1 : E_1, \ldots, y_m : E_m$, via the 'Ahn and Kolb-translation'. Rather than adding this segment directly to the context of the hearer (as in section 6.3), we propose to add it in the 'decorated' form $r_1^S : entity, \ldots, r_n^S : entity, y_1 : B_H K_H B_S K_S E_1, \ldots, y_m : B_H K_H B_S K_S E_m$. The discourse referents $r_1, \ldots, r_n$ are marked with the agent index of the speaker, to signify that the context of the hearer was extended with these referents to accommodate an utterance of the speaker. Since these referents are created on account of the speaker, the hearer should be allowed to use them in reasoning about knowledge or beliefs of the speaker by means of the $transfer_3^a$-rule. The entries, which represent the propositional content of the utterance, are prefixed with the modality $B_H K_H B_S K_S$ prescribed by the epistemic transfer rule to account for the epistemic effect of the utterance on the hearer. In the general format of the previous section the rule for adding an utterance 'U' of agent (a) to the context (Γ_b) of another agent (b) looks as follows:

AddUtt
$a : \text{'}U\text{'} \Rightarrow \Gamma_b, r_1^a : entity, \ldots, r_n^a : entity, y_1 : B_b K_b B_a K_a E_1, \ldots, y_m : B_b K_b B_a K_a E_m$
where $r_1 : entity, \ldots, r_n : entity, y_1 : E_1, \ldots, y_m : E_m$ is a type theoretical representation of the discourse U, and $r_1, \ldots, r_n, y_1, \ldots, y_m$ are Γ_b-fresh.

[4]Proof by Elias Thijsse (personal communication). The proof suggests an even stronger equivalence: $B_S K_S \varphi$ is consistent iff $\alpha_H B_S K_S \varphi$ is consistent, where α_H is any positive modality in B_H, K_H, and their duals.

Before we check this procedure, we must make sure that the rule describes a valid extension; that adding segments of this form to Γ_b will result in a correct context. The first condition that has to be fulfilled is that all variables occurring as subjects in the extended context are different. This is already guaranteed separately for Γ_b and the segment, hence the proviso that the subjects of the segment are Γ_b-fresh ensures this for the extended context. Secondly, the well-typedness conditions have to be fulfilled, i.e.:

- $\Gamma_b, r_1^a : entity, \ldots, r_i^a : entity \ \vdash \ entity : Set$ for all $i : 1 \leq i \leq n$.
- $\Gamma_b, r_1^a : entity, \ldots, r_n^a : entity, y_1 : B_b K_b B_a K_a E_1, \ldots, y_j : B_b K_b B_a K_a E_j \ \vdash$ $B_b K_b B_a K_a E_{j+1} : Prop$ (for $j : 1 \leq j < m$).

For the well-typedness of the referents all that is needed is that the type *entity*(: *Set*) is declared in the context of the hearer, but this is ensured by the Vocabulary Assumption (cf. section 6.1). By the same assumption, the context of the hearer contains all the well-typedness information needed to derive for every entry (E_{j+1}) that it is of type *Prop*, with the exception of the well-typedness of elements and proofs occurring in it. However, the only elements that can occur in E_{j+1} are the referents $r_1^a, \ldots, r_n^a$, which are declared *before* E_{j+1} in the extended context. Similarly, the only proofs that can occur in E_{j+1} are $y_1, \ldots, y_m$, the proofs of the preceding entries. Since the definition of Γ_t (section 6.1) shows that the well-typedness information needed for a modalized entry ($B_b K_b B_a K_a E_{j+1}$) is the same as for that entry without the modality (E_{j+1}), we can conclude that the second well-typedness condition is also met. Hence the Vocabulary Assumption has the intended effect of preventing that the addition of an utterance goes wrong because of a difference in vocabulary between the speaker and the hearer.

To see whether the *AddUtt*-rule makes any sense, we start by checking a simple example with respect to the inferences the hearer can make using the information he gets by adding an utterance of the speaker. Suppose that the hearer (H) knows that every farmer who owns a donkey beats it (the 'donkey-ownership rule'), $\Gamma_H \equiv \Gamma, z : (\Pi u : entity.\Pi v : entity.\Pi p_1 : farmer(u).\Pi p_2 : donkey(v).\Pi p_3 : owns(u,v).beats(u,v))$, and that the speaker ($S$) makes a speech to the effect that Pedro is a farmer, Jerry is a donkey, and that Pedro owns Jerry. Under *AddUtt* the context of the hearer will be extended with a decorated version of the corresponding segment $r_1 : entity, r_2 : entity$, $y_3 : owns(r_1, r_2)$, and become:

$$\Gamma_H \equiv \Gamma, z : (\Pi u : entity.\Pi v : entity.\Pi p_1 : farmer(u).\Pi p_2 : donkey(v).\Pi p_3 : owns(u,v) .beats(u,v)), r_1^S : entity, r_2^S : entity, y_1 : B_H K_H B_S K_S(Pedro = r_1^S), y_2 : B_H K_H B_S K_S(Jerry = r_2^S), y_3 : B_H K_H B_S K_S owns(r_1^S, r_2^S).$$

On this context, the hearer cannot in any way derive that Pedro beats Jerry: he cannot conclude that he believes this himself, since he is not convinced of the information provided by the speaker. Technically, the speaker-modality B_S in front of the entries blocks all applications of the general 'donkey-ownership rule' known by the hearer to the information about Pedro and Jerry. It is also impossible for the hearer to prove that the speaker believes that Pedro beats Jerry, since the context contains no evidence that the speaker is aware of the donkey-ownership rule. Hence in this example the *AddUtt*-rule seems cautious enough.

Although the epistemic transfer rule in [Thijsse 1992] was not intended for epistemic predicate logic, and the fragment of *DRT* covered so far does not have a construction rule for intensional verbs like 'to believe' or 'to know', we now try to find out what happens to the

Moore-sentences under *AddUtt*. As in the logical translation of these sentences, we assume that the intensional verbs are represented as modal operators, i.e. in a segment representing 'I believe that φ' the entries representing the propositional content of φ will be prefixed with the modal operator B_i (entries are now formulas in modal predicate logic, cf. $\Box PRED2$, section 5.1). If a sentence like 'φ, but I don't believe that φ' represented in this way is added to the context of the hearer using *AddUtt*, the hearer is able to derive the inconsistency using a derivation very similar to the natural deduction proof in the previous section. Besides these 'mono-logical' Moore sentences, there are also variants involving more than one person, for instance:

(4) He knows that φ but I don't believe it: $K_x\varphi \wedge \neg B_i\varphi / K_x\varphi \wedge \neg B_i K_x\varphi$.

This sentence has two readings, depending on the reference of 'it', both of which are inconsistent under *UTT* and *epistemic transfer*. It may seem less aberrant than the pragmatically anomalous sentences we have encountered before, but Thijsse remarks that a felicitous utterance of this sentence would involve an ironic intonation of 'know'[5].

To see whether this analysis carries over to the predicate logical case, we look at a variant of (4) where the 'he' is replaced by 'you' (the hearer), the first occurrence of φ by a universally quantified formula, and its second occurrence by an instantiated version of this quantified formula:

(5) You know that $\forall x.\varphi(x)$ but I don't believe $\varphi(a)$.

In line with our previous agricultural examples, we shall take $\forall x\varphi(x)$ to stand for the donkey-ownership rule and add the sentence

(6) You know that every farmer who owns a donkey beats it, but I don't believe that Pedro beats Jerry.

to the context of the hearer under *AddUtt*. Of course this can only lead to inconsistency if it is clearly understood by speaker and hearer that Pedro is a farmer, Jerry is a donkey, and that Pedro owns Jerry. Hence we assume that the speaker has said all this just before he uttered (6), and add the following segment to the context of the hearer: $r_1 : entity, r_2 : entity, y_1 : (Pedro = r_1), y_2 : (Jerry = r_2), y_3 : farmer(r_1), y_4 : donkey(r_2), y_5 : owns(r_1, r_2), y_6 : K_H(\Pi u : entity.\Pi v : entity.\Pi p_1 : farmer(u).\Pi p_2 : donkey(v).\Pi p_3 : owns(u,v).beats(u,v))$,
$y_7 : \neg B_S beats(r_1, r_2)$, where the latter two entries represent the propositional content of (6). After the application of *UttAdd* to this representation the context of the hearer looks like this:

$$\begin{aligned}
&\Gamma_H \equiv \Gamma, r_1^S : entity, r_2^S : entity,\\
&y_1 : B_H K_H B_S K_S(Pedro = r_1^S), y_2 : B_H K_H B_S K_S(Jerry = r_2^S),\\
&y_3 : B_H K_H B_S K_S K_H farmer(r_1^S), y_4 : B_H K_H B_S K_S donkey(r_2^S),\\
&y_5 : B_H K_H B_S K_S \neg B_S owns(r_1^S, r_2^S), y_6 : B_H K_H B_S K_S K_H(\Pi u : entity.\Pi v : entity.\\
&\Pi p_1 : farmer(u).\Pi p_2 : donkey(v).\Pi p_3 : owns(u,v).beats(u,v)),\\
&y_7 : B_H K_H B_S K_S \neg B_S beats(r_1^S, r_2^S).
\end{aligned}$$

On this context the inconsistency of uttering (6) can be derived in much the same way as for the utterance of (4) (in the first reading) under *epistemic transfer* in epistemic/doxastic

[5] If the irony where made explicit, the utterance would be something like 'He thinks that he knows (believes to know) that φ but I don't believe it', which is a pragmatically unproblematic sentence.

propositional logic. Since the derivation is both too long and too wide to reproduce in full, we show only the crucial middle part and use a few abbreviations. In the beginning of the derivation each of the statements added to Γ except $y_1 : B_H K_H B_S K_S(Pedro = r_1^S)$ and $y_2 : B_H K_H B_S K_S(Jerry = r_2^S)$, is brought to a '$\boxdot_H^B\ \varepsilon\ \boxdot_H^K\ \varepsilon\ \boxdot_S^B\ \varepsilon\ \boxdot_S^K$'-subordinate context by 4 subsequent applications of K-import (r_1^S and r_2^S by $transfer_3^a$). In this way the modalities are stripped from the types and a situation arises in which there is proof that r_1^S is a farmer, r_2^S is a donkey, r_1^S owns r_2^S, and that the speaker does not believe that r_1^S beats r_2^S and the hearer knows the donkey-ownership rule (lines 1-7). The proof objects M_3-M_7 are abbreviations, where $M_i \equiv \check{k}_S^K(\check{k}_S^B(\check{k}_H^K(\check{k}_H^B y_i)))$ for $i \in \{3,4,5,6,7\}$. In the derivation we omit the agent- and operator indices of the import- and export functions.

1. $\Gamma_H\ \boxdot_H^B\ \varepsilon\ \boxdot_H^K\ \varepsilon\ \boxdot_S^B\ \varepsilon\ \boxdot_S^K\ \varepsilon\ \vdash\ r_1^S : entity$
2. $\Gamma_H\ \boxdot_H^B\ \varepsilon\ \boxdot_H^K\ \varepsilon\ \boxdot_S^B\ \varepsilon\ \boxdot_S^K\ \varepsilon\ \vdash\ r_2^S : entity$
3. $\Gamma_H\ \boxdot_H^B\ \varepsilon\ \boxdot_H^K\ \varepsilon\ \boxdot_S^B\ \varepsilon\ \boxdot_S^K\ \varepsilon\ \vdash\ M_3 : farmer(r_1^S)$
4. $\Gamma_H\ \boxdot_H^B\ \varepsilon\ \boxdot_H^K\ \varepsilon\ \boxdot_S^B\ \varepsilon\ \boxdot_S^K\ \varepsilon\ \vdash\ M_4 : donkey(r_2^S)$
5. $\Gamma_H\ \boxdot_H^B\ \varepsilon\ \boxdot_H^K\ \varepsilon\ \boxdot_S^B\ \varepsilon\ \boxdot_S^K\ \varepsilon\ \vdash\ M_5 : owns(r_1^S, r_2^S)$
6. $\Gamma_H\ \boxdot_H^B\ \varepsilon\ \boxdot_H^K\ \varepsilon\ \boxdot_S^B\ \varepsilon\ \boxdot_S^K\ \varepsilon\ \vdash\ M_7 : \neg B_S beats(r_1^S, r_2^S)$
7. $\Gamma_H\ \boxdot_H^B\ \varepsilon\ \boxdot_H^K\ \varepsilon\ \boxdot_S^B\ \varepsilon\ \boxdot_S^K\ \varepsilon\ \vdash\ M_6 : K_H(\Pi u.\Pi v.\Pi p_1.\Pi p_2.\Pi p_3.beats(u,v))$
8. $\Gamma_H\ \boxdot_H^B\ \varepsilon\ \boxdot_H^K\ \varepsilon\ \boxdot_S^B\ \varepsilon\ \boxdot_S^K\ \varepsilon\ \boxdot_H^K\ \varepsilon\ \vdash\ \check{k}M_6 : (\Pi u.\Pi v.\Pi p_1.\Pi p_2.\Pi p_3.beats(u,v))$
9. $\Gamma_H\ \boxdot_H^B\ \varepsilon\ \boxdot_H^K\ \varepsilon\ \boxdot_S^B\ \varepsilon\ \boxdot_S^K\ \varepsilon\ \vdash\ \hat{t}(\check{k}M_6) : (\Pi u.\Pi v.\Pi p_1.\Pi p_2.\Pi p_3.beats(u,v))$
10. $\Gamma_H\ \boxdot_H^B\ \varepsilon\ \boxdot_H^K\ \varepsilon\ \boxdot_S^B\ \varepsilon\ \boxdot_S^K\ \varepsilon\ \vdash\ \hat{t}(\check{k}M_6)r_1^S : (\Pi v.\Pi p_1.\Pi p_2.\Pi p_3.beats(r_1^S, v))$
11. $\Gamma_H\ \boxdot_H^B\ \varepsilon\ \boxdot_H^K\ \varepsilon\ \boxdot_S^B\ \varepsilon\ \boxdot_S^K\ \varepsilon\ \vdash\ \hat{t}(\check{k}M_6)r_1^S r_2^S : (\Pi p_1.\Pi p_2.\Pi p_3.beats(r_1^S, r_2^S))$
12. $\Gamma_H\ \boxdot_H^B\ \varepsilon\ \boxdot_H^K\ \varepsilon\ \boxdot_S^B\ \varepsilon\ \boxdot_S^K\ \varepsilon\ \vdash\ \hat{t}(\check{k}M_6)r_1^S r_2^S M_3 : (\Pi p_2.\Pi p_3.beats(r_1^S, r_2^S))$
13. $\Gamma_H\ \boxdot_H^B\ \varepsilon\ \boxdot_H^K\ \varepsilon\ \boxdot_S^B\ \varepsilon\ \boxdot_S^K\ \varepsilon\ \vdash\ \hat{t}(\check{k}M_6)r_1^S r_2^S M_3 M_4 : (\Pi p_3.beats(r_1^S, r_2^S))$
14. $\Gamma_H\ \boxdot_H^B\ \varepsilon\ \boxdot_H^K\ \varepsilon\ \boxdot_S^B\ \varepsilon\ \boxdot_S^K\ \varepsilon\ \vdash\ \hat{t}(\check{k}M_6)r_1^S r_2^S M_3 M_4 M_5 : beats(r_1^S, r_2^S)$
15. $\Gamma_H\ \boxdot_H^B\ \varepsilon\ \boxdot_H^K\ \varepsilon\ \boxdot_S^B\ \varepsilon\ \vdash\ \hat{k}(\hat{t}(\check{k}M_6)r_1^S r_2^S M_3 M_4 M_5) : K_S beats(r_1^S, r_2^S)$
16. $\Gamma_H\ \boxdot_H^B\ \varepsilon\ \boxdot_H^K\ \varepsilon\ \boxdot_S^B\ \varepsilon\ \boxdot_S^K\ \varepsilon\ \vdash\ \check{4}(\hat{k}(\hat{t}(\check{k}M_6)r_1^S r_2^S M_3 M_4 M_5)) : K_S beats(r_1^S, r_2^S)$
17. $\Gamma_H\ \boxdot_H^B\ \varepsilon\ \boxdot_H^K\ \varepsilon\ \boxdot_S^B\ \varepsilon\ \boxdot_S^K\ \varepsilon\ \boxdot_S^B\ \varepsilon\ \vdash\ \check{f}(\check{4}(\hat{k}(\hat{t}(\check{k}M_6)r_1^S r_2^S M_3 M_4 M_5))) : beats(r_1^S, r_2^S)$
18. $\Gamma_H\ \boxdot_H^B\ \varepsilon\ \boxdot_H^K\ \varepsilon\ \boxdot_S^B\ \varepsilon\ \boxdot_S^K\ \varepsilon\ \vdash\ \hat{k}(\check{f}(\check{4}(\hat{k}(\hat{t}(\check{k}M_6)r_1^S r_2^S M_3 M_4 M_5)))) : B_S beats(r_1^S, r_2^S)$
19. $\Gamma_H\ \boxdot_H^B\ \varepsilon\ \boxdot_H^K\ \varepsilon\ \boxdot_S^B\ \varepsilon\ \boxdot_S^K\ \varepsilon\ \vdash\ \hat{k}(\check{f}(\check{4}(\hat{k}(\hat{t}(\check{k}M_6)r_1^S r_2^S M_3 M_4 M_5))))M_7 : \bot$

From the hearer's knowledge of the donkey-ownership rule it follows that this rule holds (line 7-9). Hence the rule can be used in combination with the information about Pedro and Jerry supplied by the speaker, to obtain a proof of $beats(r_1^S, r_2^S)$ (line 9-14). Since this is derived inside a categorical K_S-subordinate proof, it follows by positive introspection ($K_S\varphi \supset K_S K_S \varphi$) that the speaker knows that r_1^S beats r_2^S (line 14-16). Knowledge implies belief (line 16-18), and so the hearer has proof in the $\boxdot_H^B\ \varepsilon\ \boxdot_H^K\ \varepsilon\ \boxdot_S^B\ \varepsilon\ \boxdot_S^K$-subordinate context that the speaker both believes and disbelieves that r_1^S beats r_2^S, a contradiction. From this contradiction the hearer can derive a number of epistemic/doxastic conclusions ranging from $B_H K_H B_S K_S \bot$ (the hearer is convinced that the speaker is convinced of a contradiction), through $B_H K_H B_S \bot$, $B_H B_S K_S \bot$, and $B_H \bot$ to $\bot$ (the information state of the hearer is inconsistent), depending on the combination of K-, D-, and T-export used to bring it back to Γ_H.

6.6. Discussion

From the last example, it may appear that *AddUtt* forces the hearer to be too cooperative: he has to add the segment corresponding to the speakers utterance even though this makes his context inconsistent. Of course a truly cooperative speaker would not utter a Moore-sentence ((6) is also inconsistent under UTT), but the example does point out a general problem. Even adding a consistent segment to a consistent hearer-context may result in an inconsistent context. One way to deal with this would be to add the condition to *AddUtt* that the segment may not be added if the resulting context will be inconsistent. Aside from technical difficulties relating to this condition, one can imagine dialogue situations in which the hearer would prefer to add the segment and change his own (initial) context to restore consistency (for instance when the speaker is an expert on the topic of conversation). In the current $MPTS$s, a formal procedure for this 'belief revision' cannot be defined, but it should be remarked that the propositions-as-types interpretation offers interesting prospects for such a formalization.

If an agent can derive that his context is inconsistent, he will possess a proof object inhabiting '$\perp$'. This term contains the proof objects of the propositions that are jointly inconsistent: the 'culprits'. By removing these culprits from the context, this particular inconsistency will no longer be derivable. Repeating the procedure for (possible) other proofs of inconsistency will eventually result in a consistent context. In this procedure, identification and removal of culprits is straightforward, but selection is very coarse. In general the agent will want to remove a set of culprits that is 'minimal' or 'optimal' in some way, rather than simply throw all of them out of the context. However, it is questionable whether a formal procedure for belief revision should compute such a minimal choice without further information: it stands to reason that the hearer will invoke the help of the speaker if he has to revise his information state due to an utterance of the speaker.

In another respect, the update procedure does not allow the hearer to act as 'cooperative' as he should. Under *AddUtt* every utterance of the speaker is represented type theoretically via a DRS. Consequently the hearer will have to add new referents to his context with every utterance of the speaker, even if 'conversationally' no new referents have been introduced. Even though *AddUtt* was meant to capture the effect of a single utterance by the speaker, it should be possible for the hearer to identify referents 'across utterances'. In $\lambda\Box PRED2$ this identification is possible but only for referents connected to names. If a speaker mentions the same name in several utterances, the context of the hearer will contain multiple referents 'linked' to this name. Suppose that the speaker has mentioned Pedro and Jerry separately to the hearer in earlier utterances, then the context of the hearer will contain entries to this effect and look like this:

$$\Gamma_H \equiv \Gamma, r^S_{22}, y_{15} : B_H K_H B_S K_S(Pedro = r^S_{22}), \Gamma', r^S_{28}, y_{19} : B_H K_H B_S K_S(Jerry = r^S_{28}), \Gamma''$$

If the speaker now says 'Pedro owns Jerry', the hearer will add the segment r^S_{65}, $y_{33} : B_H K_H B_S K_S(Pedro = r^S_{65}), r^S_{66}, y_{34} : B_H K_H B_S K_S(Jerry = r^S_{66})$, $y_{35} : B_H K_H B_S K_S(owns(r^S_{65}, r^S_{66}))$, which results in a context that has two referents linked to Pedro (r^S_{22}, r^S_{65}) and to Jerry (r^S_{28}, r^S_{66}). However, he can derive the identity of the 'old' and the 'new' referents by the properties of the (Weak) Leibniz Identity: from $y_{15} : B_H K_H B_S K_S(Pedro = r^S_{22})$ and $y_{33} : B_H K_H B_S K_S(Pedro = r^S_{65})$, a proof object ($M_1$) can be constructed for $B_H K_H B_S K_S(r^S_{22} = r^S_{65})$. Similarly, $y_{19} : B_H K_H B_S K_S(Jerry = r^S_{28})$ and $y_{34} : B_H K_H B_S K_S(Jerry = r^S_{66})$ suffice to construct an inhabitant (M_2) of $B_H K_H B_S K_S(r^S_{28} =$

r_{66}^S). Since M_1 and M_2 prove that the old referents for Pedro and Jerry are identical to the new referents, the hearer can interpret the information provided by the last utterance of the speaker as applying to the old referents: $B_H K_H B_S K_S(owns(r_{22}^S, r_{28}^S))$.

Due to the interpretation of names as set-variables and links through Leibniz Identity, $\lambda\Box PRED2$ is able to represent the identification of referents linked to the same name. In a similar manner as for names, several referents for an indeterminate expression like 'a farmer' could be introduced into the hearer's context by a series of speaker utterances, but *MPTS*s have no formal means to express the identification of such referents. It would require the substitutions of referents from the hearer-context for referents in the segment, as well as the substitution of proofs (occurring in, or derivable on the hearers context) for the proof variables in the segment (cf. [Ahn and Kolb 1990]).

More in general, this form of anchoring is likely to be of importance in formalizing the effect of other speech acts, like questions, on the information state of the hearer. In the propositions-as-types perspective, a natural view of questions is to see them as requests for the hearer to provide (part of) a proof object for a type given by the speaker. What information the hearer has to supply about this proof object depends on the type of question: for a 'Yes/No'-question he would have to check *whether* an inhabitant of the given type can be derived on his context, whereas for a 'Why'-question, he would have to communicate the entire proof object (within the limits of what is cooperative).

Finally, a serious dialogue formalization should include a natural language representation that makes better use of the expressivity of type theory. We have used *DRT* in combination with the Ahn and Kolb-translation because it provides us with ready-made representations of the kind we need, but for all its merits, *DRT* has one important drawback: it is untyped. The universe of discourse is totally unstructured; all information about referents must be expressed via predication. If the discourse calls for the introduction of, say, a donkey, the translation will yield a segment containing the statements $r_i : entity, y_j : donkey(r_i)$, whereas type theoretically we could have expressed this more directly by means of the set-type 'donkey': $r_i : donkey$. Using the expressivity of type theory, a more direct correspondence between type theoretical representation and syntactic structure of natural language sentences can be achieved, e.g. representing nouns by set-types and adjectives as predicates over these types.

Given such a typed discourse representation, the indexing of the referents with the agent-index of the speaker by *AddUtt* is of more importance than it has been up to now. In view of the discussion of the $transfer_3$-rule in section 6.1, the reader may have expected to see examples in which the hearer instantiates a universal proposition uttered by the speaker $(B_H K_H B_S K_S \forall x.\varphi(x))$ with a referent (r_i) that is not familiar to the speaker, incorrectly obtaining a proof of $B_H K_H B_S K_S \varphi(r_i)$. However, due to the untyped referents of *DRT*, most universal quantifications in the segments are of the form $B_H K_H B_S K_S(\forall x.\varphi(x) \supset \psi(x))$, where the predicate φ selects the entities that are donkeys, farmers, etc. For such entries the hearer cannot derive that $B_H K_H B_S K_S \psi(r_i)$ without a proof of $B_H K_H B_S K_S \varphi(r_i)$, in the presence of which it would be difficult to maintain that the speaker is not familiar with r_i.

Chapter 7

Concluding remarks

In this final chapter we summarize our results, comment on some related work and indicate directions for future research.

7.1. Results

In the preface of this monograph, we argued that type theoretical contexts are suitable for the representation of information states, but that type theory suffers from 'rigidity' (all information represented in a context has the same degree of certainty) and 'loneliness' (it represents the epistemic progress of a sole agent). Our stated aim was to do something about this by incorporating intensional reasoning in type theory, through a propositions-as-types interpretation of modal logic. We have succeeded in giving a general formulation of modal type systems, and in the interpretation of various modal logics in these systems. The results are listed below, divided between the two main ingredients of the interpretation: Fitch-style natural deduction and Modal Pure Type Systems.

Fitch-style modal deduction

Fitch-style modal deduction makes use of *strict* subordinate proofs, in which only formulas of a certain form may be 'imported', and from which formulas may only be 'exported' under certain conditions. The natural deduction system for the basic modal logic K ($\Box PROP_{fitch}$, section 1.3.3) can trivially be extended to accommodate any stronger modal logic by simply allowing the additional modal axioms to be written anywhere in a natural deduction proof without further justification ('extension by axioms'). A more interesting way of strengthening the system is to vary the rules for the import and export of formulas into and out of a strict subordinate proof ('extension by rules'). For each of the standard modal axioms 4, 5, B, D, and T, a single import or export rule is given that characterizes it. These rules are used to extend the deduction system for K with any combination of 4, 5, B, D, and T in a modular way (section 1.1.4).

Although the general scope of the extension by rules is unknown (cf. the discussion in section 1.1.5), this method can successfully be applied to logics with multiple agents and multiple modalities by indexing the subordinate proofs with respect to agents and modal operators. In chapter 4, we give a natural deduction system for a logic KB_{CD} of [Kraus and Lehmann 1986]. This system combines a logic for knowledge ($KT45$) with a logic for belief ($KD45$), and the 'group modalities' 'common knowledge' ($C\varphi$) and 'common belief'

($D\varphi$). The interaction between knowledge and belief is given by the axioms $K_a\varphi \supset B_a\varphi$ and $B_a\varphi \supset K_aB_a\varphi$; similar interactions exist between common knowledge and knowledge, and between common belief and belief. Apart from indexing the rules mentioned above, the deductive characterization of this logic requires two additional import rules to deal with the interactions (section 4.1.2). The only aspect of KB_{CD} that cannot be captured by 'rules' is the introduction of the group modalities as expressed in the 'induction axioms'. This is not surprising, since both modalities are an abbreviation of an infinite conjunction of modal statements (e.g. $C\varphi =_{def} E\varphi \wedge EE\varphi \wedge EEE\varphi \wedge \ldots$, where $E\varphi$ means 'every agent knows φ'). Nevertheless, this extension to multi-agent and multi-modal deduction is satisfactory: by generalizing the additional import rules to cases with three modal operators, we find a deductive characterization of the classes of 'inspection formulas' given model theoretically in [Van der Hoek 1992].

In modal predicate logic, there is less conformity between Fitch-style deduction and model theory. The standard model theoretical account of the interaction between modality ('$\Box$') and quantification ('$\forall$') in terms of the relation between the domains of a world and its accessible worlds (cf. section 5.2.1), does not carry over to natural deduction: an extension by rules can only be found for one of the two axioms expressing this interaction. This so-called 'Converse Barcan Formula'($\Box\forall x \in D.\varphi(x) \supset \forall x \in D.\Box\varphi(x)$) becomes provable when variables that are declared in the main proof may be used in the strict subordinate proof, a rule which is already used (implicitly) in [Fitch 1952]. In the type system corresponding to the natural deduction system $\Box PRED2$ (section 5.1.1), this rule for the use of variables falls out naturally as '$transfer_3$'. The other interaction principle, the 'Barcan formula' ($\forall x \in D.\Box\varphi(x) \supset \Box\forall x \in D.\varphi(x)$), can only be dealt with through 'extension by axiom', both in the natural deduction system and the type system.

Modal Pure Type Systems

The Modal Pure Type Systems presented in this thesis are a generalization of the well-known Pure Type Systems, for which the propositions-as-types interpretation is well-understood. The 'Logic Cube' is a group of PTSs specially proposed for this purpose, in which propositional and predicate logics ranging from minimal propositional logic to higher order intuitionistic predicate logic can be interpreted. In analogy we define a 'Modal Logic Cube' (section 1.2.4), and prove that the MPTSs in this cube retain the desirable meta theoretical properties of PTSs: Unicity of Types, Subject Reduction, Strong Normalization, and Church Rosser (chapter 3). In chapter 2, we give a detailed account of the interpretation of the (second order) propositional modal logic $\Box PROP2$ in the MPTS $\lambda\Box PROP2$ along the lines of [Geuvers 1993]. Mappings are defined both from natural deduction proofs to lambda terms (the '!'-mapping), and vice versa (the '?'-mapping). We prove the soundness of these mappings (section 2.3 and 2.4 respectively), and show that the terms are invariant under composition of the mappings $((M^?)^! \equiv M)$. For the natural deduction proofs, invariance is proved for a subclass of the proofs of $\Box PROP2$, modulo some innocent duplication of formulas: for an A-OK proof Σ, $(\Sigma^!)^? \equiv_{doubles} \Sigma$. These mappings and the proofs of their properties can easily be adapted to cope with the multi-agent multi-modal logics in chapter 4. In section 5.1, we indicate what has to be done to interpret the modal predicate logic $\Box PRED2$ in $\lambda\Box PRED2$.

Since these results imply partial isomorphism between natural deduction proofs and lambda terms, 'modal' proof reductions can be formalized in $\lambda\Box PROP2$. In section 2.6, we define a number of annihilation-rules which remove certain combinations of modal functions occurring in terms (like $\check{k}\hat{k}, \check{4}\hat{\iota}$, and $\check{\iota}\hat{5}$). The effect of these rules is the removal of 'detours' (consisting of pointless combinations of import and export steps) from a modal deduction proof. In section 3.5, these annihilations are shown to be well-behaved in combination with β-reduction; Subject Reduction, Strong Normalization, and Church Rosser are proved.

In chapter 6, we did a finger exercise in the type theoretical formalization of communication to show how MPTSs fit in with existing ideas on this subject, like those in [Ahn 1992]. The rule *AddUtt* was proposed, giving a procedure for the 'update' of the information state of a hearer-agent by a declarative utterance of a speaker-agent, in the MPTS $\lambda\Box PRED2$. Although the proposed rule is too naïve, it does show that existing work in epistemic pragmatics (i.e. [Thijsse 1992]) can be brought to type theory using the developed modal framework. The discussion of this rule (section 6.6) indicates that a more serious type theoretical account of communication may require further technical development of the non-modal part of the MPTSs.

Our results show that intensional reasoning can be incorporated in type theory. The formal rigour of the propositions-as-types interpretation guarantees that intensional intuitions formalized in modal logics are transferred reliably to the MPTS, and the meta-theoretical properties ensure that reasoning in these systems is 'safe'. However, in their present formulation MPTSs may not be very practical for other purposes *than* proving meta theoretical properties. Readers that are not type theoreticians have probably already been frightened by the amount of syntax needed for even the simplest of examples in this thesis. They may rest assured: now that the theoretical foundations have been laid, all kinds of 'sugaring' (abbreviations, derived rules etc.) can be introduced to make MPTSs easier to handle for particular applications.

7.2. Related work

Naturally our work was influenced by that of many others, but since we have accounted for these influences in the pertinent places throughout the thesis, we want to use this section to comment on some recent work on the interpretation of modal logics in typed calculi.

An approach completely different from ours can be found in [De Queiroz and Gabbay 1995], where a functional Curry-Howard interpretation of modal logics in Labelled Deduction Systems[1] is proposed. The basic idea is to regard the modal operator '$\Box$' as a second order universal quantification: $\Box A \equiv \forall W.A(W)$. Different notions of modality (modal logics) are then characterised by variations in the natural deduction rule for $\forall(W)$-introduction, which translate in the Labelled Deductive Systems as conditions on lambda abstraction over world variables. These variations in 'labelling discipline' should function as the proof-theoretical counterpart to the various properties of the accessibility relation in modal model theory. The examples giving labelled versions of standard modal logics (K, KT, KD, KB, $KT4$, $KT5$) show that the labelling disciplines are not exact counterparts of the accessibility relation, e.g. for the logic KT the characteristic axiom $\Box(\Box A \supset A)$ instead of $(\Box A \supset A)$ is obtained. Still the view presented in this paper is an interesting one, it promises a uniform perspective for the treatment of a wide range of modal logics, both normal and non-normal.

In [Martini and Masini 1993], a typed λ-calculus for the positive $(\Box, \wedge, \supset)$ fragments of K, KT, $K4$, and $KT4$, is obtained by means of so-called '2-sequent calculi' for these logics. 2-sequents are a two dimensional generalization of Gentzen-sequents, allowing formulas

[1]See [Gabbay 1993].

(sequents) to occur at different 'levels'. The function of these levels is comparable to that of modal subordinate proofs in Fitch-style deduction: formulas can only change their level by means of the 'modal rules', □-introduction and □-elimination. By adding terms to the formulas in the 2-sequent calculus, Martini and Masini define a typed λ-calculus for the positive fragment of $KT4$. In this calculus, the operations on the terms are as usual for '$\supset$' (λ-abstraction and application) and '$\wedge$' (pairing and projection), and □-introduction and -elimination respectively apply functions *gen* (for generalization) and *ungen* to the terms. The □-introduction rule and *gen* correspond to K-export and $\hat{k}$ in MPTSs. The □-elimination rule and *ungen* function as K-import and $\check{k}$, 4-import and $\check{4}$, or a combination of K-import followed by T-export and $\hat{t}(\check{k}\ldots)$, dependent on the 'level-conditions'. For this calculus they prove Church-Rosser and Strong Normalization under combined β-reduction and $ungen(gen)$-reduction, which is an annihilation like $\hat{k}\check{k}$-reduction.

The only work we are aware of in which the interpretation of modal logic is approached via Fitch-style modal natural deduction, was done by Alex Simpson. In an unpublished paper (personal communication 1991), he independently proposes a propositions-as-types interpretation for the positive fragment of intuitionistic K. Given the similarity of our approaches, it is not surprising that his typed calculus looks a lot like ours, with generalized contexts, import and export functions, and equivalents of $\hat{k}\check{k}$- and $\check{k}\hat{k}$-reduction.

7.3. Directions for future research

Once two formal frameworks have been connected, like modal logic and type theory in this thesis, a lot of further work immediately suggests itself in asking in one framework (analogons of) questions living in the other. We concentrate here on two 'lines of questioning' that are of interest for type theoretical knowledge representation: weaker modal logics, and temporality.

Weaker modal logics

All modal systems that we have treated thus far are *classical* propositional or predicate logics, extended with a variety of modal rules and axioms. Modal systems based on weaker underlying logics are currently being explored in the setting of knowledge representation, giving rise to formalisms like partial modal logic ([Thijsse 1992]) and constructive partial modal logic ([Jaspars 1994]). The main motivation for this move is that the epistemic properties of agents that arise from classical modal logic are not always realistic in view of human reasoning. Although modal type theory is in some respects less sensitive to these mismatches than modal model theory, it could benefit from the richer spectrum of epistemic modalities in these weaker logics. The so-called 'substructural' modal logics are based on fragments of intuitionistic logic. Their interpretation in modal type theory would be of interest with respect to applications in linguistics, i.e. categorial grammar (see [Van Benthem 1991b], [Morill 1990], [Morill 1992]).

In MPTSs the 'logic rules' are the original PTS-rules, acting on generalized contexts. These rules can be varied independently of the import/export-rules and the transfer-rules, and it is this variation that creates the Modal Logic Cube. Further variations might allow us to interpret intuitionistic and substructural modal logics in MPTSs, as can be seen from the following overview.

Like the PTSs in the Logic Cube, MPTSs are basically intuitionistic systems: we had to add the axiom of double negation elimination to interpret the standard normal modal logics. Hence by removing the logical axiom(s) (along with the $transfer_{ax}$-rule), we should obtain modal type systems in which intuitionistic modal logics can be interpreted. Unfortunately, this is only half the story. $\lambda\Box PROP2$ without the logical axioms corresponds to the intuitionistic logic $HK\Box$ ([Božić and Došen 1984]). This logic is a true subsystem of the logic K, and has only one modal operator: '$\Box$'. In intuitionistic modal logic, the operator '$\Diamond$' can not be defined in terms of '$\Box$'. Possibility has to be dealt with independently, [Božić and Došen 1984] gives separate axiomatizations and models for an intuitionistic $\Diamond$-logic ($HK\Diamond$) and a system with both modalities ($HK\Diamond\Box$). Interpreting these logics requires a (preferably Fitch-style) natural deduction account of intuitionistic possibility, which in spite of recent developments[2] does not seem to be available at the moment.

There exist epistemically interesting intuitionistic logics in which the modal operators are duals (cf. [Jaspars 1994]). In these so-called 'Nelson logics', duality between '$\Box$' and '$\Diamond$' is restored by introducing an extra negation in intuitionistic logic, under which the operators are interdefinable. However, for these logics there are currently also no natural deduction systems that would make their interpretation straightforward.

In the PTS-framework, little work has been done on the interpretation of substructural logics. Nevertheless, there is quite a lot of work on lambda calculi and term calculi for these logics. Some of the ideas developed therein may be applicable to (M)PTSs. In [De Queiroz and Gabbay 1992] an extension of the propositions-as-types interpretation to substructural logics is proposed, based on conditions on lambda abstraction. Abstraction corresponds to $\supset$-intro, and in substructural logics there are various restrictions on this operation. For instance in relevance-logic ([Anderson and Belnap 1975]), a $\supset$-introduction may not be 'vacuous'; the hypothesis of a subordinate proof must be used in that subordinate proof before it may be discharged. The corresponding condition on lambda abstraction is that the abstracted variable must occur in the body of the abstraction. In (M)PTSs, this restriction would become a side condition on the *abstraction* rule:

$$\frac{\Gamma, x : A \;\vdash\; b : B \qquad \Gamma \;\vdash\; (\Pi x : A.B) : s}{\Gamma \;\vdash\; (\lambda x : A.b) : (\Pi x : A.B)} \text{ iff } x \in FV(b)$$

Other substructural logics can be characterized by similar restrictions on abstraction. Although a number of side conditions given in [De Queiroz and Gabbay 1992] are not particularly new to lambda calculus (for the relevance condition see [Church 1951]), their effect has yet to be studied systematically in the PTS-framework.

Another approach is taken in [Wansing 1993], where a propositions-as-types interpretation via sequent calculi is given for a whole family of substructural logics that are part of intuitionistic propositional logic. At the core of this family of logics lies a 'split' notion of implication: a distinction is made between 'left-searching' and 'right-searching' implication. These implications can only look for their argument on one side in the sequents in which they occur ($A/B, A \;\Rightarrow\; B$ but $A, A/B \;\not\Rightarrow\; B$). In the propositions-as-types interpretation, these directed implications are matched by 'left-looking' and 'right-looking' lambdas (λ^l, λ^r).

Temporality

Another direction in which the MPTS-framework should be extended has to do with temporality. In the information state of an agent reasoning about a changing domain, beliefs that were initially correct can become false, and vice versa. It is difficult to deal with such

[2] In [Simpson 1993] proof systems are given for a family of intuitionistic modal logics, but these are hybrid systems: in the proofs both formulas and relational conditions on worlds (like 'xRy') can occur as items.

(non-monotonic) phenomena without some type theoretical notion of time. The same holds for the type theoretical representation of 'tense and aspect' in natural language.

In the current MPTSs, we can interpret the simple Tense Logic of [Prior 1967]. This logic has operators G (it is always Going to be the case that), H (it always Has been the case that), P (somewhere in the Past, it was the case that), and F (somewhere in the Future, it will be the case that). It can basically be handled as a 'doubled version' of the normal modal logic K with two copies of the '$\Box$'- and '$\Diamond$'-operators (with one copy looking forward, and the other backward in time).

Clearly, this is not sufficient for the applications we are interested in. One possibility to increase the potential for temporal reasoning in MPTSs would be to try to accommodate (some of the) more sophisticated temporal operators that have been developed since [Prior 1967] (cf. [Van Benthem 1983]). A more intrinsically type theoretical approach would be to 'decorate' proof objects with temporal information, hence giving proofs a 'limited shelf life'.

Bibliography

[Ahn and Kolb 1990] Ahn, R. and Kolb, H.P., Discourse representation meets constructive mathematics. In *Papers from the second symposium on logic and language*, Kálmán, L. and Pólós, L. (eds.), Akadémia Kiadó, Budapest.

[Ahn 1992] Ahn, René, A type-theoretical approach to communication, in *Think!* 1.1, ITK Tilburg University.

[Anderson and Belnap 1975] Anderson, Alan R, and Belnap, Nuel D, *Entailment. The logic of relevance and necessity*, Princeton University Press, Princeton.

[Appelt 1985] Appelt, D.E., *Planning English sentences*. Studies in natural language processing, Cambridge University Press, Cambridge.

[Barendregt 1984] Barendregt, H.P., *The lambda calculus: its syntax and semantics*, revised edition. Studies in logic and the foundations of mathematics, North Holland.

[Barendregt and Hemerik 1990] Barendregt, Henk and Hemerik, Kees, Types in lambda calculus and programming languages. In *Proceedings of the ESOP conference*, Copenhagen.

[Barendregt 1991] Barendregt, Henk, Introduction to generalized type systems. In *Journal of functional programming* 1(2).

[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 1991a] Van Benthem, Johan, Reflections on epistemic logic. In: *Logique & Analyse* 133-134, 1991 (pp. 5-14).

[Van Benthem 1991b] Van Benthem, J., *Language in action*, Studies in logic and the foundations of mathematics (volume 130), North Holland, Amsterdam.

[Van Benthem 1983] Van Benthem, J.F.A.K., *The logic of time*, Reidel, Dordrecht.

[Van Benthem Jutting 1977] Van Benthem Jutting, L.S., *Checking Landau's 'Grundlagen' in the Automath system*. Ph.D. thesis, Eindhoven University of Technology.

[Berardi 1988] Berardi, S., *Towards a mathematical analysis of the Coquand-Huet calculus of constructions and the other systems in Barendregt's cube*, Dept. Computer Science, Carnegie-Mellon University and Dipartimento Matematica, Unversita di Torino, Italy.

[Beun 1989] Beun R.J. , *The recognition of declarative questions in dialogues*. Ph.D. thesis, Tilburg University.

[Borghuis 1992] Borghuis V.A.J. , Reasoning about knowledge of others. In *Think!* 1.1, ITK Tilburg University.

[Borghuis 1993] Borghuis, Tijn, Interpreting modal natural deduction in type theory. In *Diamonds and defaults*, De Rijke, Maarten (ed.), Kluwer, Dordrecht (pp. 67-102).

[Božić and Došen 1984] Božić, Milan and Došen, Kosta, Models for normal intuitionstic modal logics. In *Studia Logica* 3/84 (pp.218-245).

[Bull and Segerberg 1984] Bull, R.A. and Segerberg, K., Basic modal logic. In *Handbook of philosophical logic, Volume II*, Gabbay, D. and Guenther, F. (eds.), Reidel, Dordrecht (pp. 1-88).

[Bunt 1990] Bunt H.C. , DIT – Dynamic interpreation in text and dialogue. In *Second Symposium on Logic and Language*, Kálmán, L. and Pólos, L. (eds.), Akademiai Kiadó, Budapest.

[Chellas 1980] Chellas, Brian F., *Modal logic: an introduction*. Cambridge University Press, Cambridge.

[Church 1940] Church, A., A formulation of the simple theory of types. In *Journal of symbolic logic 5* (pp. 56-68).

[Church 1951] Church, A., *The calculi of lambda-conversion*, Annals of mathematics studies, N.6., Princeton University Press, Princeton.

[Constable et al. 1986] Constable, R.L. et al., *Implementing mathematics with the Nuprl Proof Development System*. Prentice-Hall.

[Coquand and Huet 1988] Coquand, Th., and Huet, G., The calculus of constructions. In *Information and Computation*, 76 (pp. 95-120).

[Curry 1934] Curry, H.B., Functionality in combinatory logic. In *Proc. Nat. Acad. Science USA* 20 (1934), (pp. 584-590).

[Dowek et al. 1991] Dowek, G., Felty, A., Herbelin, H., Huet, G., Paulin-Mohring, Ch., Werner, B., *The Coq proof assistant version 5.6, user's guide*. INRIA Rocquencourt - CNRS ENS Lyon.

[Van Eijck 1985] Eijck, Jan van, *Aspects of quantification in natural language*, Ph.D. thesis, Rijksuniversiteit Groningen.

[Fitch 1952] Fitch, Frederic Brenton, *Symbolic logic, an introduction*. The Ronald Press Company, New York.

[Fitch 1966a] Fitch, Frederic B., Natural deduction proofs for obligation. In *American philosophical quarterly 3* (pp. 27-38).

[Fitch 1966b] Fitch, Frederic B., Tree proofs in modal logic. In *Journal of symbolic logic 31*. Abstract of a paper presented at a meeting of the association for Symbolic Logic in conjunction with the American Philosophical Association at Chicago, Illinois, 29-30 April 1965 (p. 152).

[Fitting 1983] Fitting, Melvin, *Proof methods for modal and intuitionistic logics*. Reidel Publishing Company, Dordrecht.

[Fitting 1993] Fitting, Melvin, Modal logic. In *Handbook of logic in artificial intelligence and logic programming. Volume I: Logical foundations*, Gabbay, D., Hogger, C., and Robinson, J. (eds.), Oxford University Press (pp. 365-448).

[Gabbay 1993] Gabbay, D., *Labeled deductive systems*. Department of Computing, Imperial College, London (to appear with Oxford University Press).

[Gallin 1975] Gallin, D, *Intensional and higher-order modal logic with applications to Montague semantics*. North Holland, Amsterdam.

[Garson 1984] Garson, James W., Quantification in modal logic. In *Handbook of philosophical logic, Volume II*, Gabbay, D. and Guenther, F. (eds.), Reidel, Dordrecht (pp. 249-307).

[Geuvers 1988] Geuvers, Herman, *The interpretation of logics in type systems*. Master's thesis, University of Nijmegen.

[Geuvers 1993] Geuvers, Herman, *Logics and type systems*. Ph.D. thesis, University of Nijmegen.

[Geuvers and Nederhof 1991] Geuvers, Herman, and Nederhof, Mark-Jan, Modular proof of strong normalization for the calculus of constructions. In *Journal of functional programming* 1(2) (pp. 155-189).

[Girard et al. 1989] Girard, J.-Y., Lafont, Y., and Taylor, P., *Proofs and types*, Camb. Tracts in Theoretical Computer Science 7, Cambridge University Press.

[Grice 1989] Grice, Paul, *Studies in the way of words*. Harvard University Press, Cambridge.

[Hawthorn 1990] Hawthorn, John, Natural deduction in normal modal logic. In *Notre Dame journal of formal logic* 31(2).

[Hindley and Seldin 1986] Hindley, J. Roger, and Seldin, Jonathan P., *Introduction to combinators and λ-calculus*. London mathematical society student texts 1, Cambridge University Press.

[Hintikka 1962] Hintikka, J., *Knowledge and belief. An introduction to the logic of the two notions*. Cornell University Presss, Ithaca (New York).

[Van der Hoek 1992] Van der Hoek, Wiebe, *Modalities for reasoning about knowledge and quantities*, Ph.D. thesis, Free University of Amsterdam, Amsterdam.

[Howard 1980] Howard, W.A., The formulas-as-types notion of constructions. In *To H.B. Curry: Essays on combinatory logic, lambda calculus and formalism*, Seldin, J.P. and Hindley, J.R. (eds.), Academic Press, New York (pp. 479-490).

[Hughes and Cresswell 1972] Hughes, G.E. and Cresswell, M.J., *An introduction to modal logic*. University Paperback, London.

[Jaspars 1994] Jaspars, Jan, *Calculi for constructive communication: a study of the dynamics of partial states*. Ph.D. thesis, Tilburg University.

[Jones 1983] Jones, A.J.I., *Communication and meaning: an essay in applied modal logic*. Dordrecht.

[Kamp 1981] Kamp, H., A theory of truth and semantic representation. In: *Formal methods in the study of natural language*, Groenendijk, J.A.G. et al (eds.), Amsterdam.

[Kraus and Lehmann 1986] Kraus, S., and Lehmann, D., Knowledge, belief and time. In *Proceedings ICALP 1986*, Kott, L. (ed.). Lecture notes in computer science 226, Springer Verlag, Berlin.

[Kripke 1963] Kripke, S., Semantical considerations on modal logics. In *Acta philosophica fennica: Modal and many valued logics* 1963 (pp. 83-94).

[Kripke 1972] Kripke, S., Naming and necessity. In *Semantics of natural language*, Davis, S. and Harman, G. (eds.), Reidel, Dordrecht (pp. 253-355).

[Krivine and Parigot 1990] Krivine, J.-L., and Parigot, M., Programming with proofs, *J. Inf. Process. Cybern.*, EIK 26 3 (pp. 149-167).

[Lenzen 1980] Lenzen, W., *Glauben, Wissen und Wahrscheinlichkeit*, Springer Verlag, Vienna/New York.

[Luo and Pollack 1992] Luo, Zhaohui, and Pollack, Robert, *LEGO proof development system: User's manual*. Technical report ECS-LFCS-92-211, LFCS-University of Edinburgh.

[Martini and Masini 1993] Martini, Simone and Masini, Andrea, *A computational interpretation of modal proofs*. Technical report TR-27/93, Department of Computer Science, University of Pisa.

[Martin-Löf 1979] Martin-Löf, P., Constructive mathematics and computer programming. In *Logic, methodology and philosophy of science*, Vol. VI, North-Holland (pp. 153-175).

[Martin-Löf 1984] Martin-Löf, P., *Intuitionistic Type Theory*, Studies in Proof theory, Bibliopolis, Napoli.

[Meyer 1994] Meyer, J.-J.Ch., Epistemic logic for distributed systems. In *Logic: Mathematics, language, computer science and philosophy*, Vol. II, Logic and computer science, by De Swart, H.C.M. (et al.), Peter Lang, Europäischer Verlag der Wissenschaften, Frankfurt am Main (pp. 200-228).

[Morill 1990] Morill, Glyn, Intensionality and boundedness. In *Linguistics and philosophy* 13, Kluwer Academic Publishers, The Netherlands (pp. 699-726).

[Moore 1912] Moore, G,E. *Ethics*. London 1912, reprinted by Oxford University Press, Oxford 1958.

[Morill 1992] Morill, Glyn. *Categorial formalisation of relativisation: pied piping, islands and extaction sites*. Report LSI-92-23-R, Departament de llenguatges i sistemes informàtics, Universitat politècnica de Catalunya.

[Nederpelt 1977] Nederpelt, R.P., Presentation of natural deduction. In *Symposium: Set theory, foundations of mathematics*, Recueil des travaux de l'Institut Mathématique, Nouv. série, tome 2 (10). Beograd, 1977 (pp. 115-125).

[Nederpelt 1990] Nederpelt, R.P., Type systems – basic ideas and applications. In *Proceedings CSN 1990*.

[Nederpelt et al. 1994] Nederpelt, R.P., Geuvers, J.H., De Vrijer, R.C. (eds.) *Selected papers on Automath*, Studies in logic and the foundations of mathematics volume 133, North Holland.

[Paulin 1989] Paulin-Mohring, Ch., *Extraction des programmes dans le calcul des constructions*. Thèse, Université Paris VII.

[Pfennig and Paulin 1990] Pfennig, Frank, and Paulin-Mohring, Christine, Inductively defined types in the calculus of constructions. In *Mathematical foundations of programming languages*, volume 442 of *LNCS*, Springer.

[Poll 1994] Poll, E., *A programming logic based on type theory*. Thesis, Eindhoven university of Technology.

[Prior 1967] Prior, A.N.,Past, present and future, Clarendon Press, Oxford.

[De Queiroz and Gabbay 1992] De Queiroz, Ruy J.G.B. and Gabbay, Dov M., Extending the Curry-Howard interpretation to lineair, relevant and other resource logics. In *Journal of symbolic logic* 57(4) (pp. 1319-1365).

[De Queiroz and Gabbay 1995] De Queiroz, Ruy J.G.B. and Gabbay, Dov M., The functional interpretation of modal necessity. In *Advances in intensional logic* , De Rijke, Maarten (ed.). Kluwer Academic Publishers, Dordrecht.

[Ranta 1989] Ranta, Aarne, *Meaning in text.* . Manuscript, Academy of Finland and Department of Philosophy University of Stockholm.

[Ranta 1990] Ranta, Aarne, *Constructing possible worlds.* . Manuscript, Academy of Finland and Department of Philosophy University of Stockholm.

[Reynolds 1985] Reynolds, John. C., Three approaches to type structure. In *Mathematical foundations of Software Development*, Ehrig et al. eds., LNCS, Springer Verlag (pp. 97-138).

[Siemens 1977] Siemens, David F. Jr., Fitch-style rules for many modal logics. In *Notre Dame journal of formal logic* 13(4).

[Simpson 1993] Simpson, Alex,K., *The proof theory and semantics of intuitionistic modal logics*. Ph.D. thesis, Edingburgh.

[Thijsse 1992] Thijsse, Elias G.C., *Partial logic and knowledge representation*. Ph.D. thesis, Tilburg University, Eburon, Delft.

[Troelstra and Van Dalen 1988] Troelstra, A., and Van Dalen, D., *Constructivism in mathematics, an introduction, Volume I/II*, Studies in logic and the foundations of mathematics, vol. 121 and 123, North-Holland.

[Wansing 1992] Wansing, Heinrich, *Sequent calculi for normal modal logics*. Institute for Logic, Language and Information (ILLC), ILLC Prepublication Series LP-92-12, Amsterdam.

[Wansing 1993] Wansing, Heinrich, *The logic of information structures*. Lecture notes in artificial intelligence 681, Springer-Verlag, Berlin.

[Wansing 1995] Wansing, Heinrich, A full-circle theorem for simple tense logic. In *Advances in intensional logic*, De Rijke, Maarten (ed.), Kluwer Academic Publishers 1995.

[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.

Summary

Typed lambda calculi are currently finding an application in knowledge representation. Central to this application is the idea that the information state of an agent (animate or inanimate) can be modelled by means of a type theoretical context. In this view, the assertions about the world that make up an agent's information state are represented as type theoretical statements $(A : B)$, where the type (B) of a statement corresponds to an assertion of the agent and the term (A) inhabiting the type corresponds to the 'justification' or 'evidence' the agent has for this assertion. The development of an agent's information state is modelled by the sequential construction of a context.

However, type theory has basic limitations that have to be dealt with when we apply it to knowledge representation. It is too 'rigid' in the sense that all represented information is of the same certainty; we cannot, for instance, discern between things an agent 'knows' and things he merely 'believes'. Moreover, type theory is too 'lonely'; it lets us represent the evolving information state of a single agent, but not the (joint) development of the information states of a group of agents or the reasoning of agents about the information states of other agents.

In logic, these limitations have been 'overcome' by the development of modal logic. The various 'epistemic attitudes' an agent can have towards a proposition (such as knowing it, or believing it) are dealt with by extending the language with modal operators. Using this extended language, logics can be formulated which formalize the reasoning of agents about their own information state as well as the information states of others. It is the goal of this thesis to extend type theory in accordance with this approach to intensional reasoning, by providing a class of 'modal' type systems in which a 'propositions-as-types'-interpretation of these modal logics can be given.

The framework of our research is formed by the so-called Pure Type Systems (PTSs) (see [Barendregt 1992]), a large class of uniformly described type systems whose relation to non-modal logic is well-understood. We generalize a number of these systems to 'Modal Pure Type Systems' (MPTSs), by extending the PTS-language with modal operators, adding structure to the contexts, and providing rules that use this additional structure to handle modal operators (chapter 1).

To show that these systems can indeed accomodate intensional reasoning, we give a detailed account of the propositions-as-types interpretation of a family of normal modal logics in the MPTS $\lambda\Box PROP2$. Mappings are defined from the Fitch-style natural deduction proofs in these logics to terms in $\lambda\Box PROP2$ and vice versa. We prove soundness for these mappings as well as some invariance results for their composition (chapter 2). That the MPTSs themselves are well-behaved, is shown by proving that they inherit all desirable meta theoretical properties of the PTSs (chapter 3).

After settling these foundational issues for some standard logics with one agent and one modality, we demonstrate how MPTSs can be extended to deal with multiple agents and multiple (related) modalities, using the logic KB_{CD} of [Kraus and Lehmann 1986] as an example (chapter 4). In addition we investigate another strengthening: the interpretation of modal predicate logics in the MPTS $\lambda\Box PRED2$ (chapter 5). Finally, we indicate how the modal type systems could be put to work in the formalization of communication, presenting a finger exercise which brings together existing work on the type theoretical representation of natural language ([Ahn and Kolb 1990]) with existing work on epistemic pragmatics ([Thijsse 1992]) inside the MPTS-framework (chapter 6).

Samenvatting

Momenteel beginnen getypeerde λ-calculi toepassing te vinden in de kennisrepresentatie. Centraal in deze toepassing staat het idee dat de informatietoestand van een agent (mens of machine) gemodelleerd kan worden met behulp van een typentheoretische context. De asserties over de wereld waaruit de informatietoestand bestaat worden in deze optiek gerepresenteerd als typentheoretische uitspraken $(A : B)$, waarvan het type (B) corespondeert met een assertie van de agent en de bijbehorende term (A) met de „evidentie" die de agent voor deze assertie heeft. De ontwikkeling van de informatietoestand van een agent wordt opgevat als de (sequentiële) constructie van een context.

Typentheorie heeft echter een aantal inherente beperkingen die moeten worden opgeheven als we die willen toepassen voor kennisrepresentatie. Typentheorie is te 'star' in de zin dat alle gerepresenteerde informatie dezelfde zekerheidsgraad heeft. Zo is het bijvoorbeeld niet mogelijk om onderscheid te maken tussen dingen die de agent weet en dingen die hij slechts gelooft. Verder is typentheorie te „eenzaam" ; het is mogelijk om de ontwikkeling van de informatietoestand van één agent te representeren, maar niet de (gezamenlijke) ontwikkeling van de informatietoestanden van een groep van agenten, of het redeneren van agenten over de informatietoestanden van andere agenten.

In de logica zijn deze beperkingen „opgeheven" door de ontwikkeling van de modale logica. De verschillende epistemische attitudes die een agent kan hebben ten opzichte van een propositie (zoals „weten" of „ geloven"), worden uitgedrukt met behulp van modale operatoren. In de aldus uitgebreide taal kunnen logica's geformuleerd worden die het redeneren van agenten over hun eigen informatietoestand en die van anderen formaliseren. Doel van dit proefschrift is om typentheorie uit te breiden met deze benadering van intensioneel redeneren, door een klasse van „modale" getypeerde λ-calculi te ontwikkelen waarin deze modale logica's op een „propositions-as-types"-manier geïnterpreteerd kunnen worden.

Het raamwerk van ons onderzoek wordt gevormd door de zogeheten „Pure Type Systems" (PTS-en), zie [Barendregt 1992], een grote klasse van uniform te beschrijven getypeerde λ-calculi waarvan de relatie tot niet-modale logica goed begrepen is. Wij generaliseren een aantal van deze systemen tot „Modal Pure Type Systems" (MPTSs), door de taal van deze PTS-en uit te breiden met modale operatoren, hun contexten van meer structuur te voorzien en het gebruik van de operatoren te beregelen met behulp van deze extra structuur (hoofdstuk 1).

Om te laten zien dat deze systemen inderdaad intensioneel redeneren aankunnen, geven we een gedetailleerde uiteenzetting van de propositions-as-types-interpretatie van een familie van normale modale logica's in het MPTS $\lambda\Box PROP2$. Er worden afbeeldingen gedefinieerd van de natuurlijke-deductiebewijzen (in Fitch-stijl) in deze logica's naar de termen van $\lambda\Box PROP2$ en vice versa. We bewijzen de gezondheid van deze afbeeldingen alsmede enige invariantieresultaten voor hun compositie (hoofdstuk 2). Dat de MPTSen zelf nette formalismen zijn, wordt aangetoond door te bewijzen dat ze alle wenselijke meta-theoretische eigenschappen van PTSen behouden.

Nadat aldus een formeel fundament is gelegd voor een aantal standaardlogica's met één agent en één modale operator, laten we zien hoe MPTSs uitgebreid kunnen worden voor systemen met meer agenten en meer (gerelateerde) modaliteiten (hoofdstuk 4). Hierbij wordt de logica KB_{CD} uit [Kraus and Lehmann 1986] als voorbeeld gebruikt. Daarnaast onderzoeken we een andere versterking: de interpretatie van modale predikaatlogica in het MPTS $\lambda\Box PRED2$ (hoofdstuk 5). Tenslotte geven we aan hoe MPTSen gebruikt kunnen worden in het formaliseren van communicatie door, bij wijze van vingeroefening, bestaand werk op het gebied van de interpretatie van natuurlijke taal in typentheorie ([Ahn and Kolb 1990]) samen te brengen in het MPTS-raamwerk met bestaand werk op het gebied van epistemische pragmatiek ([Thijsse 1992]).

Curriculum Vitae

Tijn Borghuis werd geboren op 15 oktober 1963 in Oldenzaal. Hij behaalde in 1982 het VWO-diploma aan het Twentsch Carmellyceum aldaar. Na een jaar technische natuurkunde aan Universiteit Twente, studeerde hij van 1983 tot 1989 Wijsbegeerte: eerst aan de Rijksuniversiteit Groningen, daarna aan de Universiteit van Amsterdam. Daar slaagde hij in augustus 1989 voor zijn doctoraal-examen, met een scriptie over semantische netwerken en conditionele logica onder supervisie van prof.dr. J. van Benthem.

Van 1990 tot 1994 was hij werkzaam als AIO in het onderszoekzwaartepunt Dialoogvoering en Kennisopbouw (DenK) van het SamenwerkingsOrgaan Brabantse Universiteiten (SOBU), gestationeerd aan de Technische Universiteit Eindhoven. De resultaten van het daar verrichte onderzoek zijn neergelegd in dit proefschrift.

Sinds april 1994 werkt hij binnen het DenK-project als toegevoegd onderzoeker aan de uitbreiding van typentheorie met temporaliteit.

Current adress:
Faculty of Mathematics and Computing Science
Eindhoven University of Technology
P.O.Box 513
5600 MB Eindhoven
The Netherlands

e-mail: tijn@win.tue.nl

Stellingen

behorende bij het proefschrift
Coming to Terms with Modal Logic
van
Tijn Borghuis

1. Een „propositions-as-types"-interpretatie van modale logica in getypeerde λ-calculus via de predikaatlogische vertaling van de modale operator '$\Box$' ($\Box p = \forall v(wRv \rightarrow Pv)$) is technisch mogelijk, maar leidt tot een complex formalisme dat geen verder inzicht verschaft in modale bewijzen.

2. Tussen de in [1] beschreven klassen van 'inspection formulas' en 'trust formulas' bestaat een verband (zie sectie 4.1.2. van dit proefschrift).

    [1] Van der Hoek, Wiebe, *Modalities for reasoning about knowledge and quantities*, Ph.D. thesis, Free University of Amsterdam, Amsterdam.

3. De regel voor „epistemische overdracht", gegeven in [1], die uitdrukt dat de hoorder overtuigd is dat spreker overtuigd is van wat hij zegt, geformaliseerd als $B_h K_h B_s K_s \varphi$, garandeert tevens dat hoorder en spreker „wederzijds overtuigd" zijn van wat de spreker zegt, wat geformaliseerd kan worden als: $(B_h K_h B_s K_s)^n \varphi \wedge B_s K_s (B_h K_h B_s K_s)^m \varphi$ geldt voor alle $n, m \in I\!N$ waarbij $m \geq 0$ en $n \geq 1$ (met dank aan Elias Thijsse).

    [1] Thijsse, Elias G.C., *Partial logic and knowledge representation*. Ph.D. thesis, Tilburg University, Eburon, Delft 1992.

4. De in dit proefschrift (hoofdstuk 4) beschreven technieken voor de generalisatie van modale natuurlijke-deductiesystemen naar meer agenten en modaliteiten kunnen worden toegepast om de in [1] beschreven sequentcalculi voor normale modale logica's geschikt te maken voor systemen met meer agenten en modaliteiten. Daartoe dient men het structureel connectief "•", dat in deze calculi de rol vervult van de stricte subbewijzen in Fitch-stijl modale natuurlijke-deductiebewijzen ([2]), te indiceren naar agent respectievelijk operator.

    [1] Wansing, Heinrich, *Sequent calculi for normal modal logics*. Institute for Logic, Language and Information (ILLC), ILLC Prepublication Series LP-92-12, Amsterdam 1992.

    [2] Wansing, Heinrich, A full-circle theorem for simple tense logic. In *Advances in intensional logic*, De Rijke, Maarten (ed.), Kluwer Academic Publishers 1995.

5. Het is geen toeval dat in [1] de consequent volgehouden dubbele formulering van logica's met behulp van semantische tableaus en Fitch-stijl natuurlijke deductie ophoudt bij de modale predikaatlogica's met veranderende domeinen: er bestaat geen acceptabele natuurlijke-deductieregel voor de Barcan-formule (zie hoofdstuk 5 van dit proefschrift), terwijl er wel een tableauregel voor is.

[1] Fitting, Melvin, *Proof methods for modal and intuitionistic logics*. Reidel Publishing Company, Dordrecht 1983.

6. Of natuurlijke deductiesystemen met bewijzen in boomvorm (Prawitz-stijl) te prefereren zijn boven natuurlijke deductie systemen met lineaire bewijzen (Fitch-stijl) hangt af van de beoogde toepassing: lineaire systemen zijn geschikter voor het leveren van bewijzen *in* het systeem, boombewijzen voor het leveren van bewijzen *over* het systeem.

7. Het in [1] beschreven deontisch-modaal-temporele systeem beschrijft het verband tussen „moeten" (deontische operator O) en „kunnen" (modale operator $\Diamond$) niet fijnmazig genoeg voor het representeren van de in de praktijk gehanteerde redeneerwijze: „Als het niet kan zoals het moet, dan moet het maar zoals het kan".

   [1] Van Eck, J.A. A system of temporally relative modal and deontic predicate logic and its philosophical applications. In *Logique et Analyse* 100, 1982.

8. Daar een muziekstuk zijn voleinding vindt in de stilte na de laatste noot, dient de huidige applauspraktijk in de klassieke muziek (een staande ovatie tijdens het slotakkoord) stante pede gewijzigd te worden.

9. Uit het werk van de grote strategen Carl von Clausewitz ([1]) en Rinus Michels ([2]) volgt logischerwijs de stelling:

   Voetbal is de voortzetting van politiek met andere middelen.

   [1] Clausewitz, C. von, *Vom Kriege*, 1846.

   [2] Zie het lemma „Michels, Marinus Hendrikus Jacobus" in de *Grote Winkler Prins Encyclopedie*, 9e geheel herziene druk, Elsevier 1990.

10. Gezien het tempo waarin men in het house-genre de popgeschiedenis opnieuw doorloopt op zoek naar samples, kan het nooit lang meer duren voor de eerste plaat uitkomt die gebaseerd is op een sample van zichzelf.

11. Een ernstige omissie in gidsjes Nederlands voor op reis (zoals [1]) en cursusboeken Nederlands voor anderstaligen (zoals [2]) is dat geen aandacht wordt geschonken aan de vraag: „Wilt u zegeltjes?".

   [1] *Collins Dutch Phrase Finder*, Harper Collins 1994.

   [2] Gilbert, Lesley and Quist, Gerdi, *Teach Yourself Dutch, a complete course for beginners*, Hodder & Stoughton, London 1994.