Encoding Modal Logics in Logical Frameworks*

Arnon Avron,[□] Furio Honsell,[◇] Marino Miculan[◇] and Cristian Paravano[◇]

June 10, 1997

Abstract

We present and discuss various formalizations of Modal Logics in Logical Frameworks based on Type Theories. We consider both Hilbert- and Natural Deduction-style proof systems for representing both truth (local) and validity (global) consequence relations for various Modal Logics. We introduce several techniques for encoding the structural peculiarities of necessitation rules, in the typed λ-calculus metalanguage of the Logical Frameworks. These formalizations yield readily proof-editors for Modal Logics when implemented in Proof Development Environments, such as Coq or LEGO.

Keywords: Hilbert and Natural-Deduction proof systems for Modal Logics, Logical Frameworks, Typed λ-calculus, Proof Assistants.

Introduction

In this paper we discuss the possibility of designing *generalized Natural Deduction-style* systems for the important class of non-classical logics, consisting of *modal logics*. As definition of generalized Natural Deduction-style (ND-style), we take the one provided by *type-theoretical Logical Frameworks*, such as the Edinburgh Logical Framework, the Calculus of Inductive Constructions or Martin-Löf predicative Type Theory [15, 7, 33, 25]. These frameworks are based on the notions of *hypothetico-general judgement* [20] and the *judgements-as-types, λ-terms-as-proofs* paradigm [15]. ND-style systems in this sense are, possibly *multiple-judgement, reflective*[1] logical systems which try to incorporate the process of assuming and discharging hypotheses. This quite broad definition subsumes "standard" ND-style systems (e.g. as introduced by Gentzen and Prawitz [13, 27]); see Section 1.4 for more discussion of what are generalized ND-style systems.

The investigation carried out here can have also a significant practical aspect. In fact, Logical Frameworks (LF's) are the "logic specification" metalanguages of proof development environments (i.e. *proof editors* or, even better, *proof assistants*) in the style of [8, 19]. The systems we discuss, ultimately, are specifications (or encodings, or formalizations, or representations,...) of Modal Logics in the typed metalanguage of LF, and hence readily provide interactive proof assistants tailored to these logics.

The main challenge in encoding Modal Logics in Logical Frameworks is that of enforcing the side conditions on the application of the proper modal rules, i.e. *rules of proof* or more

*Partially supported by EC HCM Project No. CHRX-CT92.0046, *Lambda-Calcul Typé*, MURST 40%, CNR CC 94.0073.CT01.

[□]Department of Computer Science, Tel-Aviv University, Tel-Aviv, Israel. `mailto:aa@math.tau.ac.il`

[◇]Dipartimento di Matematica e Informatica, Università di Udine. Via delle Scienze 206, I-33100 Udine, Italy. `mailto:{honsell,miculan,paravano}@dimi.uniud.it`, `http://www.dimi.uniud.it/SLP`

[1]i.e., capable of referring to their own proofs.

generally "impure rules" in the sense of [2]. Such rules, in fact, cannot be applied uniformly to any set of premises, but are subject to various forms of restrictions, e.g.: the premises depend on no assumption; or depend only on assumptions of a certain shape (boxed, essentially boxed, etc.); or even, the premises have been derived only by proofs of a certain special shape (see Prawitz's *third version* of **S4**).

We introduce and study various encodings, in dependent typed λ-calculus, of Hilbert- and ND-style systems for both the consequence relations of *validity* and *truth* of **K**, **KT**, **K4**, **KT4** (**S4**), **KT45** (**S5**), **KJ1**. In particular, we extend and generalize the methodology developed in [3], by using judgements on proofs or exploiting the underlying λ-calculus structure of the metalanguage. For each encoding we state the appropriate faithfulness and adequacy theorem.

The reason for considering first Hilbert-style systems is that, in this more elementary setting *vis-à-vis* the management of assumptions, we can concentrate on the subtleties brought about by proof rules. Furthermore, it is likely that the generic user of modal systems, being more familiar with Hilbert-style systems, might be interested in proof editors supporting this style.

Some claims (or disclaimers) on our work are in order. Our objective is *not* that of extending to modal logics the "proposition-as-types", "generalized λ-terms-as-proofs" paradigm, as is the case in [21, 26]. We explore, rather, the possibility of extending to modal logics the "*judgements*-as-*dependent types*", "λ-*terms*-as-*ND-proofs*" paradigm of [15]. To this end we do not try to invent radically new deductive systems or new proof figures as in [21, 26], possibly using special extensions of the λ-calculus. These systems, albeit very interesting for the new insights that they can provide in the conceptual understanding of modality, and the conceptual meaning of the corresponding normalization procedure, are beyond the scope of this paper. These systems use "non-standard" metatheories, and hence they do not fall immediately under the formalization of Natural Deduction we consider; moreover, they are not directly amenable to an encoding in existing *general* proof assistants.

In this paper we rather try to provide natural encodings of *existing* and *classical* systems of modal logic (or very slight extensions of them). We want to produce *natural* editors, which do not force upon the user the overhead of unfamiliar, indirect encodings, or the burden of learning an altogether new system. A user of the original logic should transfer immediately to an editor, based on our encodings, his practical experience and "trade tricks". The only possible novelty, that he could experience would arise from the fact that the specification methodology of Logical Frameworks forces him to make precise and explicit all tacit conventions. Our approach therefore differs substantially from that of [21, 26], e.g. β-reductions of the λ-terms which encode proofs in our systems, do not represent steps of the proof normalization procedure, but only instantiation and application of Lemmata, i.e. the transitivity of the consequence relations. Of course, when we speak of "natural" encodings, we are well beyond what was called "naturalness" in [12], but this is the whole point of the paper: Modal Logics are *prima facie* problematic to represent in standard type-theoretic Logical Frameworks. What we show in this paper is that not only adequate type-theoretic encodings of Modal Logics are possible, but also that they provide an analysis of Modal Systems from a yet unexplored perspective.

In our view, the interest of this paper goes beyond that of merely tailoring Logical Frameworks to the peculiarities and idiosyncrasies of Modal Logics. LF's naturally suggest systems based on the natural deduction mechanism of assuming-discharging assumptions. Moreover, LF's allow to conceive systems which manipulate multiple judgements on formulæ and/or reason directly on their own proofs. Hence, some of the systems and encodings that we introduce and analyze, are interesting also from the purely logical point of view in that they suggest natural alternative presentations of Modal Logics. In particular, the ND-style systems with multiple consequence relations that we introduce are new, as far as we know.

The paper is organized as follows. In Section 1 we recall the basic syntactical and semantical definitions of Modal Logic and we present the classical Hilbert systems and the classical (together with some not so classical) ND-style systems for **K**, **KT**, **K4**, **KT4** (**S4**), **KT45** (**S5**), **KJ1**. In Section 2 we present briefly the main features and applications of Logical Frameworks. The encoding of the syntax of Modal Logic appears in Section 3. The encodings of the Hilbert-style systems and the ND-style systems in LF appear in Sections 4, and 5 respectively. In each section we discuss first systems for validity, then systems for truth; on several occasions we discuss more than one technique for implementing a given system. In Section 6 we relate formally these different tecniques. Final remarks, applications, and related work are discussed in Section 7. Proofs of theorems appear in the Appendix A.

We thank the anonymous referees for their helpful criticism.

1 Modal Logics

In this section, we briefly recall the basic notions of Modal Logics (see e.g. [18, 31]); we present Hilbert- and ND-style systems for representing truth and validity consequence relations for various modal logics.

1.1 Syntax and Semantics

The formulæ of the basic modal propositional language $\varphi \in \Phi$ are defined as follows:

$$\varphi ::= x \mid \varphi \supset \psi \mid \Box\varphi$$

where x ranges over the set of *atomic proposition*, denoted by Φ_a. The constant $\mathit{ff} \in \Phi_a$ denotes the always false proposition. Given $\varphi \in \Phi$, we denote by $\mathrm{FV}(\varphi)$ the set of *(free) atomic predicate variables*, defined as usual; the notion of FV is extended to sets of formulæ: $\mathrm{FV}(\Gamma) = \cup_{\varphi\in\Gamma}\mathrm{FV}(\varphi)$. By $\varphi[x_1, \ldots, x_n]$ we denote a formula φ such that $\mathrm{FV}(\varphi) \subseteq \{x_1, \ldots, x_n\}$; we define $\Phi_X \stackrel{\text{def}}{=} \{\varphi \in \Phi \mid \mathrm{FV}(\varphi) \subseteq X\}$. Finally, we take $\neg\varphi$ and $\Diamond\varphi$ as syntactic shorthands for $\varphi \supset \mathit{ff}$ and $\neg\Box\neg\varphi$, respectively.

Although the systems we will present are not committed to any particular semantics, for definiteness we recall the most common interpretation of modal logics, based on Kripke's frames and models. A *frame* is a pair $\mathcal{F} = \langle W, \rightarrow\rangle$ where W is the *domain* and $\rightarrow \subseteq W \times W$ is the *accessibility relation*. Elements of W are called *states*, and are denoted by s. A *model* is a triple $\mathcal{M} = \langle W, \rightarrow, \rho\rangle$ where $\langle W, \rightarrow\rangle$ is a frame, and $\rho : \Phi_a \rightarrow \mathcal{P}(W)$ is a *valuation*.

Given a formula φ, a model $\mathcal{M}$ and a state s, we define when φ is true in s ($s \models_{\mathcal{M}} \varphi$) inductively on the structure of the formula, as usual. In particular, $s \models_{\mathcal{M}} \Box\varphi \iff \forall s'.s \rightarrow s' \Rightarrow s' \models_{\mathcal{M}} \varphi$. If φ is true in every state of $\mathcal{M}$, we say that φ *is valid in* $\mathcal{M}$ ($\models_{\mathcal{M}} \varphi$).

1.2 Consequence Relations

According to [2, 24, 31], the semantic interpretation of formulæ gives rise to (at least) two (logical) consequence relations (CR's).

Definition 1.1 (Truth and Validity Consequence Relations) *Given* $\Gamma \subseteq \Phi$*,* $\varphi \in \Phi$*, and* M *class of models, we say that*

- φ is true in Γ *w.r.t.* M *(*$\Gamma \models_M \varphi$*) if* $\forall\mathcal{M} \in M.\forall s \in \mathcal{M}.s \models_{\mathcal{M}} \Gamma \Rightarrow s \models_{\mathcal{M}} \varphi$*;*

Axiom Schemata	Inference Rules
$K\ : \Box(\varphi \supset \psi) \supset (\Box\varphi \supset \Box\psi)$ $T\ : \Box\varphi \supset \varphi$ $4\ : \Box\varphi \supset \Box\Box\varphi$ $5\ : \Diamond\varphi \supset \Box\Diamond\varphi$ $J1 : \Box(\Box(\varphi \supset \Box\varphi) \supset \varphi) \supset \varphi$	NEC $\dfrac{\varphi}{\Box\varphi}$ φ does not depend on any assumption NEC' $\dfrac{\varphi}{\Box\varphi}$

Validity		Truth	
$\mathbf{K}'$	$= \mathbf{C} + K + \text{NEC'}$	$\mathbf{K}$	$= \mathbf{C} + K + \text{NEC}$
$\mathbf{KT}'$	$= \mathbf{K}' + T$	$\mathbf{KT}$	$= \mathbf{K} + T$
$\mathbf{K4}'$	$= \mathbf{K}' + 4$	$\mathbf{K4}$	$= \mathbf{K} + 4$
$\mathbf{KT4}'$	$= \mathbf{KT}' + 4$	$\mathbf{KT4}$	$= \mathbf{KT} + 4$
$\mathbf{KT45}'$	$= \mathbf{KT4}' + 5$	$\mathbf{KT45}$	$= \mathbf{KT4} + 5$
$\mathbf{KJ1}'$	$= \mathbf{K}' + J1$	$\mathbf{KJ1}$	$= \mathbf{K} + J1$

Figure 1: Axioms, rules and Hilbert-style systems for Modal Logics.

- φ is true in Γ *($\Gamma \models \varphi$) if* $\forall \mathcal{M} \forall s \in \mathcal{M}. s \models_{\mathcal{M}} \Gamma \Rightarrow s \models_{\mathcal{M}} \varphi$*;*
- φ is valid in Γ *w.r.t.* M *($\Gamma \Vdash_M \varphi$) if* $\forall \mathcal{M} \in M. \models_{\mathcal{M}} \Gamma \Rightarrow \models_{\mathcal{M}} \varphi$*;*
- φ is valid in Γ *($\Gamma \Vdash \varphi$) if* $\forall \mathcal{M}. \models_{\mathcal{M}} \Gamma \Rightarrow \models_{\mathcal{M}} \varphi$*.*

These definitions are extended straightforwardly to sets of formulæ, and subclasses of models: given M a set of models, we define $\models_M = \bigcap_{\mathcal{M} \in M} \models_{\mathcal{M}}$, $\Vdash_M = \bigcap_{\mathcal{M} \in M} \Vdash_{\mathcal{M}}$.

These CR's correspond to the *(model) global relation* and the *(model) local relation* of [31], respectively. They differ on the relevance given to assumptions: in the validity CR, formulæ of Γ are seen as *theorems*, true in every state, while in the truth CR they are *assumptions*, locally true in each state we consider. This difference is made apparent in

Theorem 1.2 ([31]) *For* $\Gamma \subseteq \Phi$, $\varphi \in \Phi$*:* $\Gamma \Vdash \varphi \iff \{\Box^n \psi \mid \psi \in \Gamma, n \in \mathbb{N}\} \models \varphi$.

Moreover, the usual "deduction theorem" ("$\Gamma, \varphi \models \psi \iff \Gamma \models \varphi \supset \psi$") holds only for the true CR's: it is easy to see that $x \Vdash \Box x$, but of course $\not\Vdash x \supset \Box x$.

1.3 Hilbert-style systems

Hilbert-style systems have been (and still are) very important tools in investigating axiomatizations of Modal Logics. Several kinds of such systems have been proposed; they differ essentially on the class of Kripke frames they axiomatize implicitly, and on the represented CR. All of them extend the following basic propositional calculus, which we denote by **C**:

$$\mathbf{C} \stackrel{\text{def}}{=} \boxed{\begin{array}{l} A_1 : \varphi \supset (\psi \supset \varphi) \\ A_2 : (\varphi \supset (\psi \supset \vartheta)) \supset ((\varphi \supset \psi) \supset (\varphi \supset \vartheta)) \\ A_3 : (\neg\psi \supset \neg\varphi) \supset ((\neg\psi \supset \varphi) \supset \psi) \end{array}} + \boxed{\text{MP}\ \dfrac{\varphi \quad \varphi \supset \psi}{\psi}}$$

In Figure 1 we list the axioms and rules schemata which can be added to **C** in order to obtain any of the modal logics we shall focus on, namely **K**, **KT**, **K4**, **KT4** (**S4**), **KT45** (**S5**), **KJ1** (In naming the systems we follow Lemmon's convention. Axiom $J1$ is known also as *Grz*, after Grzegorczyk.) We could have considered many other axioms and discussed also other modal logics. In fact, most of what appears in this paper can be readily adapted to any other system of modal logic. For ease of reading, we preferred to focus on this representative sample. Instantiation of these schemata will be denoted by subscripts; e.g., $A_{1\varphi,x\supset\varphi}$ denotes the formula $\varphi \supset ((x \supset \varphi) \supset \varphi)$.

These systems fall into two categories, depending on which CR is represented. These correspond to adopting different necessitation rules: the pure rule NEC' yields systems which are sound and complete only w.r.t. the validity CR's. If we are interested in the truth CR's, we need the impure rule NEC. Hilbert type systems are not always taken with assumptions, however, having in mind implementations of Hilbert systems, we take them into account from the very beginning. In fact, the possibility of managing assumptions and using previously proved Lemmata (i.e., applying the deduction theorem), is central in the process of proof development.

The basic concept of a proof of a formula φ in a Hilbert-type system S is that of a labelled *tree*. The labels are formulæ, and the formula which labels a node which is not a leaf should follow from the formulæ which label its successors by one of the rules of S. A formula φ follows in S from a set of formulæ Δ (written $\Delta \vdash_S \varphi$) iff there is a proof-tree π (of the kind just described) in which every leaf is labelled by an axiom of S or by an element of Δ, and the root is labelled by φ. (In Δ, therefore, there may be formulæ which do not label any leaf of π.) This is denoted by $\pi : \Delta \vdash_S \varphi$; the set of free variables in π is denoted by $\mathrm{FV}(\pi)$.

Definition 1.3 (Valid Proofs) *Given $X \subseteq \Phi_a, \Delta \subseteq \Phi_X, \varphi \in \Phi_X$ we say that π is a* valid proof (in the system S) of φ w.r.t. (X, Δ) *(denoted by $(X, \Delta) \models_S \pi : \varphi$) if $\pi : \Delta \vdash_S \varphi$ and $\mathrm{FV}(\pi) \subseteq X$.*

Theorem 1.4 (Completeness of Hilbert-style systems) *For $\Gamma \subseteq \Phi,\ \varphi \in \Phi$:*

1. *For $S \in \{\mathbf{K}, \mathbf{KT}, \mathbf{K4}, \mathbf{KT4}, \mathbf{KT45}, \mathbf{KJ1}\} : \Gamma \vdash_S \varphi \iff \Gamma \models_{M(S)} \varphi$;*
2. *For $S \in \{\mathbf{K}', \mathbf{KT}', \mathbf{K4}', \mathbf{KT4}', \mathbf{KT45}', \mathbf{KJ1}'\} : \Gamma \vdash_S \varphi \iff \Gamma \Vdash_{M(S)} \varphi$;*

where $M(S)$ denotes the class of models corresponding to the axioms characterizing S.

1.4 Natural Deduction-style systems

In this subsection we introduce ND-style systems for both validity and truth CR's. All these systems extend the usual ND-style system for propositional classic logic **NC** [27]:

$$\mathbf{NC} \stackrel{\mathrm{def}}{=} \boxed{\begin{array}{llll} \Gamma, \varphi \vdash \varphi & \textsc{Weak}\ \dfrac{\Gamma \vdash \varphi}{\Gamma, \Delta \vdash \varphi} & & \textsc{Raa}\ \dfrac{\Gamma, \neg\varphi \vdash \mathit{ff}}{\Gamma \vdash \varphi} \\ \supset\text{-I}\ \dfrac{\Gamma, \varphi \vdash \psi}{\Gamma \vdash \varphi \supset \psi} & & \supset\text{-E}\ \dfrac{\Gamma \vdash \varphi \supset \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \psi} & \\ \mathit{ff}\text{-I}\ \dfrac{\Gamma \vdash \varphi \quad \Gamma \vdash \neg\varphi}{\Gamma \vdash \mathit{ff}} & \mathit{ff}\text{-E}\ \dfrac{\Gamma \vdash \mathit{ff}}{\Gamma \vdash \varphi} & & \end{array}}$$

We make extensive use of ND-style systems with multiple consequence relations. We do not give here a detailed presentation of them, because we feel that their "working" is self-evident; see [24, Chapter 3] for further details. We need them in order to capture systems for validity for logics weaker than S4. Moreover, they allow to achieve a sharpening of the adequacy theorems appearing in [3]. In Section 5.2.5 we briefly outline how to introduce multiple CR systems for truth, extending those for validity. All the systems for truth appearing elsewhere in the paper are classical.

$\Box$-I $\dfrac{\Box\Gamma \vdash \varphi}{\Box\Gamma \vdash \Box\varphi}$	$\supset_\Box$-E $\dfrac{\Gamma \vdash \Box(\varphi \supset \psi) \quad \Gamma \vdash \Box\varphi}{\Gamma \vdash \Box\psi}$	$\Box$-E $\dfrac{\Gamma \vdash \Box\varphi}{\Gamma \vdash \varphi}$
$\Box'$-I $\dfrac{\emptyset \vdash \varphi}{\emptyset \vdash \Box\varphi}$	$\supset'$-E $\dfrac{\Gamma \Vdash \varphi \supset \psi \quad \Gamma \Vdash \varphi}{\Gamma \Vdash \psi}$	$\Box_\Box$-I $\dfrac{\Gamma \vdash \Box\varphi}{\Gamma \vdash \Box\Box\varphi}$
$\Box''$-I $\dfrac{\Gamma \vdash \varphi}{\Gamma \Vdash \Box\varphi}$	$\supset''$-E $\dfrac{\Gamma \vdash \varphi \supset \psi \quad \Gamma \Vdash \varphi}{\Gamma \Vdash \psi}$	$\Box_\Diamond$-I $\dfrac{\Gamma \vdash \Diamond\varphi}{\Gamma \vdash \Box\Diamond\varphi}$
$\Box'''$-I $\dfrac{\Gamma \Vdash \varphi}{\Gamma \Vdash \Box\varphi}$	$\supset'''$-E $\dfrac{\Gamma \Vdash \varphi \supset \psi \quad \Gamma \vdash \varphi}{\Gamma \Vdash \psi}$	$\Box_\supset$-E $\dfrac{\Gamma \vdash \Box(\Box(\varphi \supset \Box\varphi) \supset \varphi)}{\Gamma \vdash \varphi}$

Validity		Truth	
NK′	$=$ **NC** $+ \supset_\Box$-E $+ \Box''$-I $+ \Box'''$-I $+ \supset'$-E $+ \supset''$-E $+ \supset'''$-E	**NS4**	$=$ **NC** $+ \Box$-I $+ \Box$-E
		NK	$=$ **NC** $+ \supset_\Box$-E $+ \Box'$-I
NKT′	$=$ **NK′** $+ \Box$-E	**NKT**	$=$ **NK** $+ \Box$-E
NK4′	$=$ **NK′** $+ \Box_\Box$-I	**NK4**	$=$ **NK** $+ \Box_\Box$-I
NKT4′	$=$ **NKT′** $+ \Box_\Box$-I	**NKT4**	$=$ **NKT** $+ \Box_\Box$-I
NKT45′	$=$ **NKT4′** $+ \Box_\Diamond$-I	**NKT45**	$=$ **NKT4** $+ \Box_\Diamond$-I
NKJ1′	$=$ **NK′** $+ \Box_\supset$-E	**NKJ1**	$=$ **NK** $+ \Box_\supset$-E

Figure 2: Rules and ND-style systems for Modal Logics.

Systems are displayed in a linearized sequent-like fashion. We denote by $\pi : \Gamma \vdash^i_S \varphi$ the proof π of the fact that φ is entailed by the assumptions Γ, accordingly to the i-th CR of the system S.

In Figure 2 we display the rules which can be added to **NC** in order to obtain ND-style versions of the Modal Logics **K**, **KT**, **K4**, **KT4** (**S4**), **KT45** (**S5**), **KJ1**. In naming these systems we extend Lemmon's convention for Hilbert-style systems.

These systems count as ND-style systems, in that their rules follow the general schema

$$\forall \Gamma_1, \ldots, \Gamma_n \quad \frac{\Gamma_1, \Delta_1 \vdash^{i_1} \varphi_1 \quad \ldots \quad \Gamma_n, \Delta_n \vdash^{i_n} \varphi_n}{\cup_{i=1}^n \Gamma_i \vdash^i \varphi} C$$

where C is a possible *side condition*, that is a restriction on the applicability of the schemata, and $i, i_1, \ldots, i_n \in \{1, \ldots, m\}$ where $\vdash^1, \ldots, \vdash^m$ are the m CR of the system S. In this view, ND-style systems are characterized by the fact that one does not focus only on theorems but rather on assumption-conclusion dependencies. Rules are monotone with respect to sets of assumptions and possibly exploit assumption-discharging mechanisms. Notice that the structural rule of weakening is assumed at the outset.

Some of the readers may object that we use a non-standard definition of "natural-deduction style systems" and that our definition is liberal enough that systems in either Hilbert style or in "two-sided sequent style" (i.e. with left and right rules) would qualify. The last claim is false, since the definition above does *not* allow the introduction of new formulæ on the l.h.s. which were not already there in at least one of the premises. The possibility of doing so is the *main* characteristic of *real* two-sided sequent systems. In Natural Deduction systems, in contrast, all the "activity" is done on the r.h.s.! Our definition indeed implies that Hilbert-type systems are in principle a special case of Natural Deduction systems. After all, in the very first natural deduction formulation of classical logic [13], an *axiom* is used (namely, the *excluded middle axiom*). The difference between Hilbert-type systems and *good* natural deduction systems is in the spirit, it is not a formal one (except, of course, that Hilbert-type systems do not allow rules in which some assumption is discharged). The spirit of Natural Deduction systems is to use succinct, natural *rules*, in which certain connectives are either introduced or eliminated.[2] We believe that this is the case with our systems. They are not as close to the ideal as, say, the natural deduction system of intuitionistic logic, but this seems to be forced by the nature of modal logic. Another possible objection may be that we use rules which involve many connectives, and so do not fit the form of either an introduction or an elimination rule. Again, this complaint is based on a confusion between the essence of an introduction rule and properties of an ideal such rule. Having introduction rules which involve more than one connective is a very common, unavoidable phenomenon. Thus, in usual natural deduction systems for 3-valued logics, rules that involve a combination of negation with some other connective are standard (see, e.g., that given in [4] for an extension of Kleene's three-valued logic).

The systems in Figure 2 fall into two categories, depending on which CR is represented. **NS4**, **NK**,..., **NJ1** represent the truth CR's while **NK**$'$,..., **NKJ1**$'$ represent the validity CR's. ND-style systems are best suited to represent the truth consequence relation, since the $\supset$-I rule wraps up the deduction theorem in the system. Prawitz' system **NS4** is a good example of how to take full advantage of this [27].

On the other hand, ND-style systems for validity are cumbersome: since the deduction theorem does not hold for $\Vdash$, we can no longer adopt the usual introduction rule for implication. A possible solution for overcoming this problem appears in the system **NK**$'$ that we introduce here. This system uses two different CR's, i.e. $\vdash, \Vdash$, whose intended meaning is:

- $\Gamma \vdash \varphi$ iff "there is a proof of φ from Γ which does not use the $\Box''$-I, $\Box'''$-I rules" (these derivations are said *box-intro free*);
- $\Gamma \Vdash \varphi$ iff "there is a proof of φ from Γ which does use the $\Box''$-I, $\Box'''$-I rules".

Box-intro free proofs can be used in deriving valid consequences, but not the converse. The connection between these two notions of derivation is clear in the box introduction rules: we can "box" a valid formula still obtaining a valid formula (rule $\Box'''$-I), but if we "box" a formula obtained on the $\vdash$ level, we obtain a valid formula ($\Box''$-I). The rules $\supset'$-E, $\supset''$-E, $\supset'''$-E allow for the "modus ponens" between valid and box-intro free derived formulæ. The rule Embed $\dfrac{\Gamma \vdash \varphi}{\Gamma \Vdash \varphi}$ is however derivable:

$$\supset''\text{-E}\dfrac{\Box'''\text{-I}\dfrac{\Gamma \vdash \mathit{true}}{\Gamma \Vdash \Box \mathit{true}} \qquad \supset\text{-I}\dfrac{\text{Weak}\dfrac{\Gamma \vdash \varphi}{\Gamma, \Box \mathit{true} \vdash \varphi}}{\Gamma \vdash \Box \mathit{true} \supset \varphi}}{\Gamma \Vdash \varphi}$$

where *true* denotes any propositional tautology, e.g. $\varphi \supset \varphi$ (its derivation is omitted).

The rule $\supset_\Box$-E corresponds to the K axiom of Hilbert-style systems. The other rules for $\Vdash$ ($\supset'$-E, $\Box''$-I) correspond to the modus ponens and the necessitation rules, respectively. Rules corresponding to the axioms of the extensions of **NK**$'$, are added at the level of $\vdash$.

Of course, instead of introducing rules $\Box$-E, $\Box_\Box$-I, $\Box_\Diamond$-I, $\Box_\supset$-E, we could have postulated directly the corresponding axioms. These two choices are completely equivalent; our rules are perhaps more "natural" in view of proof search.

Notation for proofs and free variables of proofs are the same of Hibert-style systems.

[2] Gentzen introduced Natural Deduction as "a formalism that reflects as accurately as possible the actual logical reasoning involved in mathematical proofs" [13, Section 2.1].

Theorem 1.5 (Completeness of ND-style systems) *For* $\Gamma \subseteq \Phi,\ \varphi \in \Phi$*:*

1. *For* $S \in \{\mathbf{NK}, \mathbf{NKT}, \mathbf{NK4}, \mathbf{NKT4}, \mathbf{NKT45}, \mathbf{NKJ1}, \mathbf{NS4}\} : \Gamma \vdash_S \varphi \iff \Gamma \models_{M(S)} \varphi$*;*

2. *For* $S \in \{\mathbf{NK}', \mathbf{NKT}', \mathbf{NK4}', \mathbf{NKT4}', \mathbf{NKT45}', \mathbf{NKJ1}'\}$*:*

   (a) $\Gamma \Vdash_S \varphi \iff \Gamma \models\!\!\models_{M(S)} \varphi$

   (b) $\Gamma \vdash_S \varphi \iff \Gamma \models_{M_*} \varphi$

where $M(S)$ *denotes the class of models corresponding to the rules characterizing* S*, and* M_* *is the class of models on the trivial frame* $\mathcal{F}_* \stackrel{\text{def}}{=} \langle\{*\}, \{(*, *)\}\rangle$*.*

Proof. (Sketch) 1., 2.a: Each axiom and rule is easily proved to be sound. Completeness is proved by deriving a suitable complete Hilbert-style system found in literature.

2.b: In every model of the trivial frame $\mathcal{F}_*$, the formulæ $\Box\varphi, \Diamond\varphi$ are equivalent to φ, because the only world $*$ is also its only own successor. Therefore, we can erase the $\Box$'s and $\Diamond$'s from formulæ and rules, obtaining equivalent formulæ and rules which are purely propositional. Hence, the $\vdash$ rules of a system S are adequate with respect to M_* iff their "erased" versions are adequate wrt the class of propositional models. This is easily proved to hold for every $S \in \{\mathbf{NK}', \mathbf{NKT}', \mathbf{NK4}', \mathbf{NKT4}', \mathbf{NKT45}', \mathbf{NKJ1}'\}$. $\Box$

2 Logical Frameworks

Type Theories, such as the Edinburgh Logical Framework [15, 3] or the Calculus of Inductive Constructions [7, 33] were especially designed, or can be fruitfully used, as a general logic specification language, i.e. as a Logical Framework. In an LF, we can represent faithfully and uniformly all the relevant concepts of the inference process in a logical system: syntactic categories, terms, assertions, axiom schemata, rule schemata, tactics, etc. via the "judgements-as-types λ-terms-as-proofs" paradigm. The key concept is that of *hypothetico-general* judgement [20], which is rendered as a type of the dependent typed λ-calculus of the Logical Framework. The λ-calculus metalanguage of an LF supports *higher order* syntax. *Substitution*, α-conversion of bound variables and *instantiation of schemata* are also taken care of uniformly by the metalanguage. Since LF's allow for higher order assertions (*judgements*) one can treat on a par axioms and rules, theorems and derived rules, and hence encode also generalized natural deduction systems in the sense of [28].

Encodings in LF's often provide the "normative" formalization of logic under consideration. The specification methodology of LF's, in fact, forces the user to make precise all tacit, or informal, conventions, which always accompany any presentation of a logic.

Any interactive proof development environment for the type theoretic metalanguage of an LF (e.g. Coq [8], LEGO [19]), can be readily turned into one for a specific logic. We need only to fix a suitable environment (the *signature*), i.e. a declaration of typed constants corresponding to the syntactic categories, term constructors, judgements, and rule schemata. Such an LF-generated editor allows the user to reason "under assumptions" and go about in developing a proof the way mathematicians normally reason: using hypotheses, formulating conjectures, storing and retrieving lemmata, often in top-down, goal-directed fashion. It is worth noticing that the LF feature of supporting reasoning under assumptions necessarily gives a ND-style flavour to any encoding of a logic in LF.

LF provide a common medium for integrating different systems. Hence LF-derived editors rival special purpose editors when efficiency can be increased by integrating independent logical systems. LF-generated editors are *natural.* A user of the original logic can transfer immediately to them his practical experience and "trade tricks." They do not force upon the user the overhead of unfamiliar indirect encodings, as would editors, say derived from FOL editors, via an encoding.

The wide conceptual universe provided by LF allows, on various occasions, to device genuinely new presentations of the logics. This will be the case for some of the encodings for Modal Logics in this paper. In particular, we shall capitalize on the feature of LF's of treating simultaneously different judgements and of treating proofs as first-class objects.

In this paper, we work in the Edinburgh Logical Framework, as presented in [15].

3 Encoding of the Syntax

In encoding the language of Modal Logic we follow the LF paradigm [15, Section 3]: the syntactic category Φ is represented by the type o of propositions; for each syntactic constructor, we introduce a corresponding constructor over o. Propositional variables $(x, y, \ldots)$, are directly represented by metalogical variables of LF $(\mathrm{x}, \mathrm{y}, \ldots)$. The signature $\Sigma(\Phi)$ for the language and the encoding function $\varepsilon_X : \Phi_X \to o$ appears below:

$$\Sigma(\Phi) = \boxed{\begin{array}{lcl} o & : & Type \\ ff & : & o \\ \Box & : & o \to o \\ \supset & : & o \to o \to o \end{array}} \qquad \boxed{\begin{array}{rcl} \varepsilon_X(x) & \stackrel{\text{def}}{=} & \mathrm{x} \;\text{ if } x \in X \\ \varepsilon_X(ff) & \stackrel{\text{def}}{=} & ff \\ \varepsilon_X(\Box\varphi) & \stackrel{\text{def}}{=} & \Box\varepsilon_X(\varphi) \\ \varepsilon_X(\varphi \supset \psi) & \stackrel{\text{def}}{=} & \supset\; \varepsilon_X(\varphi)\; \varepsilon_X(\psi) \end{array}}$$

Given a set $X = \{x_1, \ldots, x_n\}$ of propositional variables, we denote by Γ_X the context $\langle \mathrm{x}_1 : o, \ldots, \mathrm{x}_n : o\rangle$.

Theorem 3.1 *Given $X \subseteq \Phi_a$, the function ε_X is a compositional bijection between Φ_X and the canonical forms*[3] *of type o in $\Sigma(\Phi), \Gamma_X$. Moreover, the encoding is compositional in the sense that for $X = \{x_1, \ldots, x_n\}, Y \subseteq \Phi_a$, $\varphi \in \Phi_X$ and $\varphi_1, \ldots, \varphi_n \in \Phi_Y$: $\varepsilon_Y(\varphi[x_1 := \varphi_1, \ldots, x_n := \varphi_n]) = \varepsilon_X(\varphi)[\mathrm{x}_1 := \varepsilon_Y(\varphi_1), \ldots, \mathrm{x}_n := \varepsilon_Y(\varphi_n)]$.*

All the systems we shall deal with have the same language. Hence, the signatures, that we will introduce in the rest of the paper, will include $\Sigma(\Phi)$ without explicit mention.

4 Encodings of Hilbert-style systems

4.1 Systems for validity

The encodings of these systems follow the LF paradigm for specifying a logical system [15, Section 4]. In Figure 3 we give the signature $\Sigma(\mathbf{K}')$ for the Hilbert-style system $\mathbf{K}$, and its extensions for other systems ($\mathbf{K4}'$, ...).

Given $\Delta \subseteq \Phi_X$, we define the LF context $\gamma_V(\Delta)$ as follows:

$$\gamma_V(\Delta) \stackrel{\text{def}}{=} \begin{cases} \langle\rangle & \text{if } \Delta \equiv \emptyset \\ \gamma_V(\Delta'), v : (V\ \varepsilon_X(\varphi)) & \text{if } \Delta \equiv \Delta', \varphi \text{ and } v \text{ fresh for } \gamma_V(\Delta') \end{cases}$$

Henceforth, as a syntactic shorthand, we will denote by v_φ the (unique) variable v such that $(v{:}(V\ \varepsilon_X(\varphi))) \in \gamma_V(\Delta)$, for $\varphi \in \Delta$. We will adopt this notation also in later encodings (where, of course, the involved judgement may be different from V).

[3] The notion of *canonical form* is very close to that of long $\beta\eta$-normal form; see [15] for details.

Judgements			
V	:	$o \to Type$	
Axioms and Rules			
A_1	:	$\prod_{x,y:o}(V\ \varepsilon_{\{x,y\}}(A_{1x,y}))$	Similarly for $A_{2x,y,z}, A_{3x,y}, K_{x,y}$.
MP	:	$\prod_{x,y:o}(V\ x) \to (V(\supset xy)) \to (V\ x)$,	
NEC	:	$\prod_{x:o}(V\ x) \to (V\ (\Box x))$	
4	:	$\prod_{x:o}(V\ \varepsilon_{\{x\}}(4_x))$	Similarly for $T_x, 5_x, J1_x$.

Figure 3: $\Sigma(\mathbf{K'})$ and its extensions for $\mathbf{K4'}, \ldots$.

We can then define the *encoding function* $\varepsilon^{\Sigma(\mathbf{K'})}_{X,\Delta}$, where $X \subseteq \Phi_a,\ \Delta \subseteq \Phi_X$; such function maps proofs π of $\mathbf{K'}$ such that $\mathrm{FV}(\pi) \subseteq X$ to canonical forms of type $(V\ \varepsilon_X(\varphi))$, for $\varphi \in \Phi_X$, in the environment $\Sigma(\mathbf{K'}), \Gamma_X, \gamma_V(\Delta)$:

$\varepsilon^{\Sigma(\mathbf{K'})}_{X,\Delta} : \{\pi \mid (X,\Delta) \models_{\mathbf{K'}} \pi : \varphi, \varphi \in \Phi_X\} \to \{t \mid \Gamma_X, \gamma_V(\Delta) \vdash_{\Sigma(\mathbf{K'})} t : (V\ \varepsilon_X(\varphi)), \varphi \in \Phi_X\}$	
$\varepsilon^{\Sigma(\mathbf{K'})}_{X,\Delta}(\varphi) \stackrel{\text{def}}{=} v_\varphi$	if $\varphi \in \Delta$
$\varepsilon^{\Sigma(\mathbf{K'})}_{X,\Delta}(A_{1\varphi,\psi}) \stackrel{\text{def}}{=} A_1\ \varepsilon_X(\varphi)\ \varepsilon_X(\psi)$	similarly for A_2, A_3, K
$\varepsilon^{\Sigma(\mathbf{K'})}_{X,\Delta}\ (\mathrm{NEC}_\varphi(\pi)) \stackrel{\text{def}}{=} NEC\ \varepsilon_X(\varphi)\ \varepsilon^{\Sigma(\mathbf{K'})}_{X,\Delta}(\pi)$	
$\varepsilon^{\Sigma(\mathbf{K'})}_{X,\Delta}\ (\mathrm{MP}_{\varphi,\psi}(\pi,\pi')) \stackrel{\text{def}}{=} MP\ \varepsilon_X(\varphi)\ \varepsilon_X(\psi)\ \varepsilon^{\Sigma(\mathbf{K'})}_{X,\Delta}(\pi)\ \varepsilon^{\Sigma(\mathbf{K'})}_{X,\Delta}(\pi')$	

Theorem 4.1 *The function $\varepsilon^{\Sigma(\mathbf{K'})}_{X,\Delta}$ is a compositional bijection between proofs π, such that $(X,\Delta) \models_{\mathbf{K'}} \pi : \varphi$, and canonical terms p,*[4] *such that $\Gamma_X, \gamma_V(\Delta) \vdash_{\Sigma(\mathbf{K'})} p : (V\ \varepsilon_X(\varphi))$.*

4.2 Systems for truth

In encoding these systems, we have to deal with the problematic issue of enforcing the side condition of the necessitation rule. Hence, we have to extend accordingly the LF methodology for encoding assertions. Here we consider three solutions. In the first, we add a new parameter to the basic judgement, i.e. $T : U \to o \to Type$, where U is a type on which no constructor is defined. In the second, we introduce a new judgement on proof terms, corresponding to the metatheoretic notion that "the proof depends on no assumption." The third solution makes use of two judgements over formulæ, $Ta, V : o \to Type$. It follows closely the one in [3, Section 4.1]. In Section 6 we shall elaborate on the connection between these three solutions.

4.2.1 World parameters

In Figure 4 we give the signature $\Sigma_w(\mathbf{K})$ for the Hilbert-style system $\mathbf{K}$, and its extensions for other systems ($\mathbf{K4}$, $\mathbf{KT}$, ...). The extra sort U (the *universe*) has no constructors: therefore, the only terms inhabiting U are variables, which have to be assumed in the typing context. These variables are called suggestively "worlds" (of the universe). It should be noticed, however, that this terminology is chosen only for its intuitive appeal, and there is no direct connection with Kripke semantics of modal logics. Indeed, we *do not* introduce any

[4] In the following, we denote generic terms by t, proof forms by p, proofs of no-assumption judgement by n, proofs of closed judgement by c,...

Syntactic Categories		
U	:	*Type*
Judgements		
T	:	$U \to o \to Type$
Axioms and Rules		
A_1	:	$\prod_{x,y:o} \prod_{w:U} (T\ w\ (\varepsilon_{\{x,y\}}(A_{1x,y}))$ Similarly for $A_{2x,y,z}, A_{3x,y}, K_{x,y}$
MP	:	$\prod_{x,y:o} \prod_{w:U} (T\ w\ \mathrm{x}) \to (T\ w\ (\supset \mathrm{xy})) \to (T\ w\ y)$
NEC	:	$\prod_{x:o} (\prod_{w:U} (T\ w\ \mathrm{x})) \to \prod_{w:U} (T\ w\ (\Box \mathrm{x}))$
4	:	$\prod_{x:o} \prod_{w:U} (T\ w\ \varepsilon_{\{x\}}(4_x))$ Similarly for $T_x, 5_x, J1_x$.

Figure 4: $\Sigma_w(\mathbf{K})$, and its extensions for **K4**,...

accessibility relation (as it is done in semantic embeddings, e.g. [5]). Hence, this approach is general enough to allow to encode easily any *proof rule*, also in Logics whose semantics does not rely upon Kripke frames (such as, e.g., Linear Logic; see also [17, 24] for applications to Dynamic Logic and Hoare Logic).

The idea behind the use of the extra world parameter is purely syntactical. By means of this extra parameter we succeed in representing the side condition of "no assumptions" in proof rules, in terms of the metalogical condition of "no free variables" in proof terms. In making an assumption, we are forced to assume the existence of a world, say w, and to instantiate the truth judgement T also on w. This judgement appears also as an hypothesis on w. Hence, deriving as premise a judgement, which is universally quantified with respect to U, amounts to establishing the judgement for a generic world on which no assumptions are made, i.e. on no assumptions.

The encoding function $\varepsilon^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}$ is inductively defined on the structure of proofs: given a proof $\pi : \Delta \vdash_{\mathbf{K}} \varphi$, $\varepsilon^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(\pi)$ is the proof term corresponding to π, where $X = \mathrm{FV}(\pi)$.

$$\begin{array}{l}
\varepsilon^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(\varphi) \stackrel{\mathrm{def}}{=} v_\varphi \qquad if\ \varphi \in \Delta \\
\varepsilon^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(A_{1,\varphi\psi}) \stackrel{\mathrm{def}}{=} A_1\ \varepsilon_X(\varphi)\ \varepsilon_X(\psi)\ w \qquad \text{similarly for } A_2, A_3, K \\
\varepsilon^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(\mathrm{N{\small EC}}_\varphi(\pi)) \stackrel{\mathrm{def}}{=} NEC\ \varepsilon_X(\varphi)\ (\lambda w'{:}U.\varepsilon^{\Sigma_w(\mathbf{K})}_{X,\emptyset,w'}(\pi))\ w \\
\varepsilon^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(\mathrm{MP}_{\varphi\psi}(\pi,\pi')) \stackrel{\mathrm{def}}{=} MP\ \varepsilon_X(\varphi)\ \varepsilon_X(\psi)\ w\ \varepsilon^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(\pi)\ \varepsilon^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(\pi')
\end{array}$$

Given a variable w of type U, $\Delta \subseteq \Phi$ with $\mathrm{FV}(\Delta) \subseteq X$, we define the LF context $\gamma_w(\Delta)$ as follows:

$$\gamma_w(\Delta) \stackrel{\mathrm{def}}{=} \begin{cases} w : U & \text{if } \Delta \equiv \emptyset \\ \gamma_w(\Delta'), v : (T\ w\ \varepsilon_X(\varphi)) & \text{if } \Delta \equiv \Delta', \varphi \text{ and } v \text{ fresh for } \gamma_w(\Delta') \end{cases}$$

Theorem 4.2 *The function $\varepsilon^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}$ is a compositional bijection between proofs π, such that $(X,\Delta) \models_{\mathbf{K}} \pi : \varphi$, and canonical terms t, such that $\Gamma_X, \gamma_w(\Delta) \vdash_{\Sigma_w(\mathbf{K})} t : (T\ w\ \varepsilon_X(\varphi))$.*

4.2.2 "No Assumptions"-judgement

In Figure 5 we give the signature $\Sigma_{Na}(\mathbf{K})$ and its extensions for the systems **K4**, **KT**, ...

Given $\Delta \subseteq \Phi$ with $\mathrm{FV}(\Delta) \subseteq X$, we define the LF context $\gamma_T(\Delta)$ as follows:

$$\gamma_T(\Delta) \stackrel{\mathrm{def}}{=} \begin{cases} \langle\rangle & \text{if } \Delta \equiv \emptyset \\ \gamma_T(\Delta'), v : (T\ \varepsilon_X(\varphi)) & \text{if } \Delta \equiv \Delta', \varphi \text{ and } v \text{ fresh for } \gamma_T(\Delta') \end{cases}$$

Judgements

$T{:}o \to Type$

$Na{:}\prod_{x:o} Tx \to Type$

Axioms and Rules

$A_1{:}\prod_{x,y:o}(T\ (\varepsilon_{\{x,y\}}(A_{1x,y})))$ Similarly for $A_{2x,y,z}, A_{3x,y}, K_{x,y}$

$MP{:}\prod_{x,y:o}(T\ (\supset xy)) \to (T\ x) \to (T\ y)$

$NEC{:}\prod_{x:o}\prod_{d:(T\ x)}(Na\ x\ d) \to (T\ \Box x)$

$Na_{A_1}{:}\prod_{x,y:o}(Na\ \varepsilon_{\{x,y\}}(A_{1x,y})\ (A_1\ x\ y))$, Similarly for $A_{2x,y,z}, A_{3x,y}, K_{x,y}$

$Na_{NEC}{:}\prod_{x:o}\prod_{d:(T\ x)}\prod_{n:(Na\ x\ d)}(Na\ \Box x\ (NEC\ x\ d\ n))$

$Na_{MP}{:}\prod_{x,y:o}\prod_{d_1:(T\ x)}\prod_{d_2:(T\ (\supset xy))}(Na\ x\ d_1) \to (Na\ (\supset xy)\ d_2) \to (Na\ y\ (MP\ x\ y\ d_2\ d_1))$

$4{:}\prod_{x:o}(T\ \varepsilon_{\{x\}}(4_x))$

$Na_4{:}\prod_{x:o}(Na\ \varepsilon_{\{x\}}(4_x)\ (4\ x))$ Similarly for $T_x, 5_x, J1_x$

Figure 5: $\Sigma_{Na}(\mathbf{K})$ and its extensions for **K4**,. . . .

The adequacy theorem relies on two technical lemmata (the second is in Section A.2.4):

Lemma 4.3 *$\forall t, p$ canonical forms: $\Gamma_X \vdash_{\Sigma_{Na}(\mathbf{K})} p{:}(T\ t) \Rightarrow \exists n.\Gamma_X \vdash_{\Sigma_{Na}(\mathbf{K})} n{:}(Na\ p\ t)$.*

Following the steps of the proof of Lemma 4.3, it is easy to define a function α which maps each canonical form p, such that $\Gamma_X \vdash_{\Sigma_{Na}(\mathbf{K})} p : (T\ t)$ to the corresponding proof term n such that $\Gamma_X \vdash_{\Sigma_{Na}(\mathbf{K})} n : (Na\ p\ t)$. Then we can define the encoding function for $\Sigma_{Na}(\mathbf{K})$ as follows[5]:

$$\begin{array}{ll}
\varepsilon^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}(\varphi) \stackrel{\text{def}}{=} v_\varphi & if\ \varphi \in \Delta \\
\varepsilon^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}(A_{1\varphi\psi}) \stackrel{\text{def}}{=} A_1\ \varepsilon_X(\varphi)\ \varepsilon_X(\psi) & similarly\ for\ A_{2\varphi,\psi,\vartheta}, A_{3\varphi,\psi}, K_{\varphi,\psi} \\
\varepsilon^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}(\mathrm{NEC}_\varphi(\pi)) \stackrel{\text{def}}{=} NEC\ \varepsilon_X(\varphi)\varepsilon^{\Sigma_{Na}(\mathbf{K})}_{X,\emptyset}(\pi)\ \alpha\left(\varepsilon^{\Sigma_{Na}(\mathbf{K})}_{X,\emptyset}(\pi)\right) & \\
\varepsilon^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}(\mathrm{MP}_{\varphi,\psi}(\pi,\pi')) \stackrel{\text{def}}{=} MP\ \varepsilon_X(\varphi)\ \varepsilon_X(\psi)\ \varepsilon^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}(\pi)\ \varepsilon^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}(\pi') &
\end{array}$$

Theorem 4.4 *The function $\varepsilon^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}$ is a compositional bijection between valid proofs π, such that $(X, \Delta) \models_{\mathbf{K}} \pi : \varphi$, and canonical terms p, such that $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Na}(\mathbf{K})} p : (T\ \varepsilon_X(\varphi))$.*

Notice that, in order to use the above signature faithfully to the original system, the user should not assume any Na assertion.

4.2.3 Two-judgements systems

We next describe a method in which the two consequence relations, validity and truth, are handled together, in one comprehensive system. The method is rather general, and can be used for every Hilbert-type system in which the rules are divided into rules of derivation and rules of proof.

We start with the following observation. Recall that a proof of a formula φ in a Hilbert-type system $\mathcal{H}$ is a labelled *tree*. A formula φ follows in $\mathcal{H}$ from a set of formulæ Δ (written $\Delta \vdash_{\mathcal{H}} \varphi$) iff there is a proof-tree π in which every leaf is labelled by an axiom of $\mathcal{H}$ or by an element of Δ, the root is labelled by φ, and the following condition is be satisfied:

[5]Recall that v_φ is the unique variable v such that $(v{:}(T\ \varepsilon_X(\varphi))) \in \gamma_T(\Delta)$.

- The formula which labels a node which is not a leaf should follow from the formulæ which label its successors by one of the rules of $\mathcal{H}$.

Now the main property of a *pure* Hilbert-type system is that for such a system the condition above has a *local* character. By this we mean that all we need to know in order to check it at a certain node, are the formulæ which label that node and its successors. This is *not* the case, e.g. if one of the rule is a rule of proof. Checking validity of a node which is justified by such a rule requires (among other things) checking the leaves of all the branches which pass through that node and see that they all are labelled by axioms. This is a *global* condition on the subtree of which that node is the root!

The solution to this problem is to arrange things so that all the data which is needed for checking validity of a node would be found at that node and its successors. For rules of proof this can be achieved rather easily by adding to each node a second label. This second label is either the word *valid* or the word *true*. Officially, therefore, each node is labelled by a pair $\langle\varphi, l\rangle$, where φ is a formula and $l \in\{true, valid\}$. Let us call a tree π of such pairs a *generalized $\mathcal{H}$-proof* of $\langle\varphi, l\rangle$ from the set of assumptions $\langle\varphi_1, l_1\rangle, \ldots, \langle\varphi_n, l_n\rangle$, (written $\pi : \langle\varphi_1, l_1\rangle, \ldots, \langle\varphi_n, l_n\rangle \vdash_{\mathcal{H}'} \langle\varphi, l\rangle$) if the following conditions are satisfied:

- As a tree of formulæ, the tree is a legitimate proof-tree of the system $\mathcal{H}'$, which is obtained from $\mathcal{H}$ by turning any rule of proof into a rule of derivation.
- A leaf $\langle\psi, h\rangle$ has $h = valid$ if ψ is an axiom of $\mathcal{H}$.
- A node which is not a leaf is labelled *valid* iff all its successors are labelled *valid*.
- A node which is derived by a rule of proof of $\mathcal{H}$ should be labelled *valid* (hence so should also be the case for every node in the subtree which is generated by it).

It is a straightforward task now to prove the following

Lemma 4.5 *The erasing of the second label is a compositional bijection between:*

1. *proofs in $\mathcal{H}'$ and generalized $\mathcal{H}$-proofs, in which all nodes are labelled* valid.
2. *(ordinary) proofs in $\mathcal{H}$ and generalized $\mathcal{H}$-proofs, in which all leaves which are not labelled by axioms are labelled* true.

It is obvious, therefore, that generalized $\mathcal{H}$-proofs subsume ordinary proofs in both $\mathcal{H}$ and $\mathcal{H}'$. On the other hand they behave nicely from the LF point of view, and so can easily be represented. One possibility is to view generalized $\mathcal{H}$-proofs as ordinary proofs of a pure Hilbert-type system of *signed* formulæ (where the signs are *true* and *valid*). An equivalent approach which is perhaps more intuitive is to introduce *two* judgements, "T" (for "truth") and "V" (for "validity"). The corresponding obvious representation in the case of the modal logics treated above is given in Figure 6.

Theorem 4.6 *There is a compositional bijection between generalized $\mathcal{H}$-proofs (where $\mathcal{H} =$ **K**, **K4**, etc.) of $\langle\varphi_1, l_1\rangle, \ldots, \langle\varphi_n, l_n\rangle \vdash_{\mathcal{H}} \langle\psi, l\rangle$ and canonical terms t such that*

$$\Gamma_X, \gamma_V(\{\varphi_i \mid l_i = valid\}), \gamma_T(\{\varphi_i \mid l_i = true\}) \vdash_{\Sigma_{2j}(\mathcal{H})} t : (J\ \varepsilon_X(\psi))$$

where $J = T$ if $l =$ true, V otherwise.

Corollary 4.7 *Suppose $\{\varphi_1, \ldots, \varphi_n, \psi\} \subseteq \Phi_X$.*

Judgments			
T, V	:	$o \to Type$	
Axioms and Rules			
A_1	:	$\prod_{x,y:o}(V\ \varepsilon_{\{x,y\}}(A_{1x,y}))$	Similarly for $A_{2x,y,z}, A_{3x,y}, K_{x,y}$
$MP_{T,T}$	:	$\prod_{x,y:o}(T\ (\supset xy)) \to (T\ x) \to (T\ y)$	
$MP_{V,V}$	:	$\prod_{x,y:o}(V\ (\supset xy)) \to (V\ x) \to (V\ y)$	
$MP_{T,V}$	:	$\prod_{x,y:o}(T\ (\supset xy)) \to (V\ x) \to (T\ y)$	
$MP_{V,T}$	:	$\prod_{x,y:o}(V\ (\supset xy)) \to (T\ x) \to (T\ y)$	
NEC	:	$\prod_{x:o}(V\ x) \to (V\ (\Box x))$	
4	:	$\prod_{x:o}(V\ \varepsilon_{\{x\}}(4_x))$	Similarly for $T_x, 5_x, J1_x$.

Figure 6: $\Sigma_{2j}(\mathbf{K})$ and its extensions for **K4**,...

1. *There is a compositional bijection between proofs in $\mathcal{H}'$ (where $\mathcal{H} = \mathbf{K}, \mathbf{K4}$, etc.) of $\varphi_1 \ldots \varphi_n \vdash_{\mathcal{H}} \psi$ and canonical terms t such that $\Gamma_X, \gamma_V(\{\varphi_1, \ldots, \varphi_n\}) \vdash_{\Sigma_{2j}(\mathcal{H})} t : (V\ \varepsilon_X(\psi))$.*
2. *There is a compositional bijection between proofs in $\mathcal{H}$ (where $\mathcal{H} = \mathbf{K}, \mathbf{K4}$, etc.) of $\varphi_1 \ldots \varphi_n \vdash_{\mathcal{H}} \psi$ and canonical terms t such that $\Gamma_X, \gamma_T(\{\varphi_1, \ldots, \varphi_n\}) \vdash_{\Sigma_{2j}(\mathcal{H})} t : (J\ \psi)$, where J is V if $dom(\gamma_T(\{\varphi_1, \ldots, \varphi_n\})) \cap \mathrm{FV}(t) = \emptyset$, T otherwise.*

The last corollary is nice, but it is obvious that generalized $\mathcal{H}$-proofs define, in fact, something which is stronger than both $\mathcal{H}$ and $\mathcal{H}'$. What naturally corresponds to them is a sort of a *triple* consequence relation, that is, we write $\Delta; \Xi \vdash_{\mathcal{H}} \varphi$ iff there is a generalized $\mathcal{H}$-proof in which the root is labelled by φ, while every leaf is either labelled by an axiom, or by an element of Δ and *valid*, or by an element of Ξ and *true*. This is the case, it should be emphasized, for any Hilbert-type system of the kind we treat here. In the case of modal logics, however, this triple consequence relation has a clear semantic interpretation (and has already been used, e.g., in [9], where it is denoted like this: $\Delta \models_{\mathcal{H}} \Xi \longrightarrow \varphi$):

$$\Delta; \Xi \vdash_{\mathcal{H}} \varphi \iff \forall \mathcal{M} \in M. \forall s \in \mathcal{M}. (\models_{\mathcal{M}} \Delta \wedge s \models_{\mathcal{M}} \Xi) \Rightarrow s \models_{\mathcal{M}} \varphi$$

It is clear that what we have constructed is a representation of this triple consequence relation. It is easy, in fact, to show the following generalization of the previous corollary:

Theorem 4.8 *There is a compositional bijection between generalized $\mathcal{H}$-proofs of $\Delta; \Xi \vdash_{\mathcal{H}} \varphi$ and canonical terms t such that $\Gamma_X, \gamma_V(\Delta), \gamma_T(\Xi) \vdash_{\Sigma_{2j}(\mathcal{H})} t : (J\ \varepsilon_X(\varphi))$ where J is V if $dom(\gamma_T(\Xi)) \cap \mathrm{FV}(t) = \emptyset$, T otherwise.*

Remark. In our representation the MP rule has been represented by four constants, each with a different type. In general, a rule of derivation R with n premises will be represented by 2^n constants (while a rule of proof will need just one). We can, in fact, represent any such rule by just two ($R_{V,\ldots,V}$ and $R_{T,\ldots,T}$), provided we introduce the following extra global constant:

$$C : \prod_{x:o}(V\ x) \to (T\ x)$$

Using this constant we can *define*, e.g., $MP_{T,V}$ and $MP_{V,T}$ as follows:

$$MP_{T,V} \stackrel{\text{def}}{=} \lambda x, y : o.\lambda t : (T\ (\supset xy)).\lambda s : (V\ x).(MP_{T,T}\ t\ (C\ s))$$
$$MP_{V,T} \stackrel{\text{def}}{=} \lambda x, y : o.\lambda t : (T\ (\supset xy)).\lambda s : (V\ x).(MP_{T,T}\ (C\ t)\ s)$$

Judgements		
Ta, V	:	$o \to Type$
Rules		
$\supset$-I	:	$\prod_{x,y:o}((Ta\ x) \to (Ta\ y)) \to (Ta(\supset\ x\ \psi))$
$\Box_{Ta}$-I	:	$\prod_{x:o}(Ta\ x) \to (V\ \Box x)$
$\supset$-$E_{Ta,Ta}$	:	$\prod_{x,y:o}(Ta(\supset x\ y)) \to (Ta\ x) \to (Ta\ y)$
$\Box_V$-I	:	$\prod_{x:o}(V\ x) \to (V\ \Box x)$
$\supset$-$E_{V,Ta}$	:	$\prod_{x,y:o}(V\ (\supset x\ y)) \to (Ta\ x) \to (V\ y)$
$\supset$-$E_{Ta,V}$	:	$\prod_{x,y:o}(Ta\ (\supset x\ y)) \to (V\ x) \to (V\ y)$
$\supset$-$E_{V,V}$	:	$\prod_{x,y:o}(V\ (\supset x\ y)) \to (V\ x) \to (V\ y)$
$\supset_\Box$-E	:	$\prod_{x,y:o}(Ta\ \Box(\supset\ x\ y)) \to (Ta\ \Box x) \to (Ta\ \Box y)$
$\Box$-E	:	$\prod_{x:o}(Ta\ \Box x) \to (Ta\ x)$
$\Box_\Box$-I	:	$\prod_{x:o}(Ta\ \Box x) \to (Ta\ \Box\Box x)$
$\Box_\Diamond$-I	:	$\prod_{x:o}(Ta\ \Diamond x) \to (Ta\ \Box\Diamond x)$
$\Box_\supset$-E	:	$\prod_{x:o}(Ta\ \Box(\supset \Box(\supset x\ \Box x)\ x) \to (Ta\ x)$

Figure 7: $\Sigma_{2j}(\mathbf{NK'})$ and its extensions for $\mathbf{NK4'}$,...

Similar treatment can be given to any rule of derivation. This approach has the advantage that we can require J (in Corollary 4.7 and Theorem 4.8) to be simply T, which is rather intuitive. The disadvantage is that we lose the bijection between proofs and terms: there is some amount of freedom concerning where to apply C, and so more than one term corresponds to a given proof. This can be remedied, e.g., by requiring that in canonical terms C will be applied as late as possible.

5 Encodings of Natural Deduction-style systems

Throughout this section, we shall encode only the "minimal" fragment of the modal logics. It should be straightforward to extend the signatures to the full systems.

5.1 Systems for validity

We use an extension of the two-judgements technique seen above. In Figure 7 we give the signature $\Sigma_{2j}(\mathbf{NK'})$ and its extension for systems $\mathbf{NK4'}$, $\mathbf{NKT'}$, ...

Given $\Delta \subseteq \Phi$ with $\mathrm{FV}(\Delta) \subseteq X$, we define the LF context $\gamma_{Ta}(\Delta)$ as follows:

$$\gamma_{Ta}(\Delta) \stackrel{\text{def}}{=} \begin{cases} \langle\rangle & \text{if } \Delta \equiv \emptyset \\ \gamma_{Ta}(\Delta'), v : (Ta\ \varepsilon_X(\varphi)) & \text{if } \Delta \equiv \Delta', \varphi \text{ and } v \text{ fresh for } \gamma_{Ta}(\Delta') \end{cases}$$

Theorem 5.1 *For $X \subset \Phi_a$, $\Delta \subseteq \Phi_X$, $\varphi \in \Phi_X$:*

- *There exists a compositional bijection between proofs π, such that $(X, \Delta) \models_{\mathbf{NK'}} \pi : \varphi$, and canonical terms p, such that $\Gamma_X, \gamma_{Ta}(\Delta) \vdash_{\Sigma_{2j}(\mathbf{NK'})} p : (Ta\ \varepsilon_X(\varphi))$.*

- *There exists a compositional bijection between proofs π, such that $(X, \Delta) \Vdash_{\mathbf{NK'}} \pi : \varphi$, and canonical terms p, such that $\Gamma_X, \gamma_{Ta}(\Delta) \vdash_{\Sigma_{2j}(\mathbf{NK'})} p : (V\ \varepsilon_X(\varphi))$.*

Special system for NS4. We can get an alternative Natural Deduction-style system $\mathbf{NS4'}$ for $\mathbf{NKT4'}$, closer in spirit to Prawitz' first system for S4 [27], by replacing $\supset_\Box$-E and $\Box_\Box$-I by the rule

$$\supset_\Box\text{-I}\ \frac{\Gamma, \Box\varphi \Vdash \psi}{\Gamma \Vdash \Box\varphi \supset \psi}$$

The resulting system is $\mathbf{NS4'} \stackrel{\text{def}}{=} \mathbf{NC} + \supset_\Box\text{-I} + \Box\text{-E}$. In this system, $\supset_\Box$ and $\Box_\Box$-I are derivable on the level of $\Vdash$, not $\vdash$.

The encoding of system $\mathbf{NS4'}$ is straightforward, and we get a compositional bijection. This is an improvement of the encoding used in [3, Section 4.2].

One can get an analogue of Prawitz' second system for S4 (see [27]) by using the rule

$$\supset_{EM}\text{-I}\ \frac{\Gamma, \varphi \Vdash \psi}{\Gamma \Vdash \Box\varphi \supset \psi}\ \varphi \text{ is essentially-modal}$$

The side condition can be handled, like in [3], by introducing a special judgement, $EM : o \to \mathit{Type}$, which corresponds to the property of being "essentially modal" (see [27] for definitions).

5.2 Systems for truth

We present two general solutions for handling the necessitation rule in the classical systems presented in Section 1.4: the first is based on *world parameters*, the second makes use of a *"closed assumption"-judgement*. The signatures obtained using these two approaches allow to derive the corresponding ones introduced for the Hilbert-style case in Section 4.2, and hence they are strictly stronger. In Section 5.2.5 we sketch also yet another general solution which makes use of three judgements on formulæ. Strictly speaking, this is an encoding of novel multiple CR systems for the truth CR of Modal Logics.

For the special system **NS4** introduced by Prawitz [27], we consider two more encodings. These adopt an auxiliary judgement on proofs for enforcing Prawitz's conditions ("boxed assumptions" and "boxed-fringe", respectively). Also in this section, we restrict ourselves to the "minimal" fragment of modal logic.

5.2.1 World parameters

In Figure 8 we give the signature $\Sigma_w(\mathbf{NK})$ and its extensions for the other systems (**NK4**, ...). The encoding function $\varepsilon^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w}$ is defined on the structure of proofs of **NK**: given a proof $\pi : \Delta \vdash_{\mathbf{NK}} \varphi$, $\varepsilon^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w}(\pi)$ is the proof term corresponding to π.

$$\begin{array}{l}
\varepsilon^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w}(\varphi) \stackrel{\text{def}}{=} v_\varphi \qquad \mathit{if}\ \varphi \in \Delta \\
\varepsilon^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w}(\Box'\text{-I}_\varphi(\pi')) \stackrel{\text{def}}{=} \Box\text{-I}\ \varepsilon_X(\varphi)\ (\lambda w' : U.\varepsilon^{\Sigma_w(\mathbf{NK})}_{X,\emptyset,w'}(\pi'))\ w \\
\varepsilon^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w}(\supset\text{-I}_{\varphi\psi}(\pi')) \stackrel{\text{def}}{=} \supset\text{-I}\ \varepsilon_X(\varphi)\ \varepsilon_X(\psi)\ w(\lambda v{:}(T\ w\ \varepsilon_X(\varphi)).\varepsilon^{\Sigma_w(\mathbf{NK})}_{X,(\Delta,\varphi),w}(\pi')) \\
\varepsilon^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w}(\supset\text{-E}_{\varphi\psi}(\pi',\pi'')) \stackrel{\text{def}}{=} \supset\text{-E}\ \varepsilon_X(\varphi)\ \varepsilon_X(\psi)\ w\ \varepsilon^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w}(\pi')\ \varepsilon^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w}(\pi'') \\
\varepsilon^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w}(\supset_\Box\text{-E}_{\varphi\psi}(\pi',\pi'')) \stackrel{\text{def}}{=} \supset_\Box\text{-E}\ \varepsilon_X(\varphi)\ \varepsilon_X(\psi)\ w\ \varepsilon^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w}(\pi')\ \varepsilon^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w}(\pi'')
\end{array}$$

Theorem 5.2 *The function $\varepsilon^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w}$ is a compositional bijection between proofs π, such that $(X,\Delta) \models_{\mathbf{NK}} \pi : \varphi$, and canonical terms t, such that $\Gamma_X, \gamma_w(\Delta) \vdash_{\Sigma_w(\mathbf{NK})} t : (T\ w\ \varepsilon_X(\varphi))$.*

Syntactic Categories		
U	:	*Type*
Judgements		
T	:	$U \to o \to Type$
Axioms and Rules		
$\supset$-I	:	$\prod_{x,y:o} \prod_{w:U}((T\ w\ x) \to (T\ w\ y)) \to (T\ w\ (\supset xy))$
$\supset$-E	:	$\prod_{x,y:o} \prod_{w:U}(T\ w\ (\supset xy)) \to (T\ w\ x) \to (T\ w\ y)$
$\supset_\Box$-E	:	$\prod_{x,y:o} \prod_{w:U}(T\ w\ \Box(\supset xy)) \to (T\ w\ \Box x) \to (T\ w\ \Box y)$
$\Box'$-I	:	$\prod_{x:o} (\prod_{w:U}(T\ w\ x)) \to \prod_{w:U}(T\ w\ (\Box x))$
$\Box$-E	:	$\prod_{x:o} \prod_{w:U}(T\ w\ \Box x) \to (T\ w\ x)$
$\Box_\Box$-I	:	$\prod_{x:o} \prod_{w:U}(T\ w\ \Box x) \to (T\ w\ \Box\Box x)$
$\Box_\Diamond$-I	:	$\prod_{x:o} \prod_{w:U}(T\ w\ \Diamond x) \to (T\ w\ \Box\Diamond x)$
$\Box_\supset$-E	:	$\prod_{x:o} \prod_{w:U}(T\ w\ \Box(\supset \Box(\supset x\ \Box x)\ x) \to (T\ w\ x)$

Figure 8: $\Sigma_w(\mathbf{NK})$ and its extensions for **NK4**,...

5.2.2 "Closed Assumptions"-judgement

In Figure 9 we give the signature $\Sigma_{Cl}(\mathbf{NK})$ and its extensions for the other truth systems (**NK4**, **NKT**, ...). Notice that there is a rule for establishing the "closed assumption"-judgement corresponding to each proof constructor, i.e. for each rule in **NK**.

The existence and definition of the encoding function relies upon two technical lemmata:

Lemma 5.3 *$\forall p$ canonical form, if $\Gamma_X, \Delta \vdash_{\Sigma_{Cl}(\mathbf{NK})} p : (T\ t)$ then $\exists c.\Gamma_X, \Delta, \Xi_p(\Delta) \vdash_{\Sigma_{Cl}(\mathbf{NK})} c : (Cl\ t\ p)$, where $\Xi_p(\Delta) \stackrel{\text{def}}{=} \{c{:}(Cl\ t\ x) \mid x \in \mathrm{FV}(p) \wedge (x{:}(T\ t)) \in \Delta\}$.*

Lemma 5.3 defines naturally a function from canonical proof forms $p : (T\ t)$ to canonical forms of type $(Cl\ t\ p)$, in the same environment expanded with the "closed assumptions" for the free variables of p. Let us denote such function by α.

Lemma 5.4 *$\forall c$ canonical form, if $\Gamma_X, \Delta, \Xi \vdash_{\Sigma_{Cl}(\mathbf{NK})} c{:}(Cl\ t\ p)$ then $\Gamma_X, \Delta', \Xi \vdash_{\Sigma_{Cl}(\mathbf{NK})} c{:}(Cl\ t\ p)$, where Ξ contains all and only the Cl assertions, and $\Delta' = \{x{:}(T\ t) | (Cl\ t\ x) \in \Im(\Xi)\}$.*

We can now define the encoding function $\varepsilon^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta}$, which relies on the α above mentioned.

$$\begin{array}{l}
\varepsilon^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta}(\varphi) \stackrel{\text{def}}{=} v_\varphi \qquad\qquad\qquad , if\ \varphi \in \Delta \\
\varepsilon^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta}(\supset\text{-I}_{\varphi,\psi}(\pi)) \stackrel{\text{def}}{=} \supset\text{-I}\ \varepsilon_X(\varphi)\ \varepsilon_X(\psi)\ (\lambda v{:}(T\ \varepsilon_X(\varphi)).\varepsilon^{\Sigma_{Cl}(\mathbf{NK})}_{X,(\Delta,\varphi)}(\pi)) \\
\varepsilon^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta}(\supset\text{-E}_{\varphi,\psi}(\pi',\pi'')) \stackrel{\text{def}}{=} \supset\text{-E}\ \varepsilon_X(\varphi)\ \varepsilon_X(\psi)\ \varepsilon^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta}(\pi')\ \varepsilon^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta}(\pi'') \\
\varepsilon^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta}(\supset_\Box\text{-E}_{\varphi,\psi}(\pi',\pi'')) \stackrel{\text{def}}{=} \supset_\Box\text{-E}\ \varepsilon_X(\varphi)\ \varepsilon_X(\psi)\ \varepsilon^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta}(\pi')\ \varepsilon^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta}(\pi'') \\
\varepsilon^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta}(\Box'\text{-I}_{\varphi}(\pi)) \stackrel{\text{def}}{=} \Box'\text{-I}\ \varepsilon_X(\varphi)\ \varepsilon^{\Sigma_{Cl}(\mathbf{NK})}_{X,\emptyset}(\pi)\ \alpha\left(\varepsilon^{\Sigma_{Cl}(\mathbf{NK})}_{X,\emptyset}(\pi)\right)
\end{array}$$

Theorem 5.5 *The function $\varepsilon^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta}$ is a compositional bijection between proofs π, such that $(X, \Delta) \models_{\mathbf{NK}} \pi{:}\varphi$, and canonical terms t, such that $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Cl}(\mathbf{NK})} t{:}(T\ \varepsilon_X(\varphi))$.*

Judgements

$T{:}o \to Type$

$Cl{:}\prod_{x:o}(T\ x) \to Type$

Axioms and Rules

$\supset_{\Box}\text{-E}{:}\prod_{x,y:o}(T\ \Box(\supset xy)) \to (T\ \Box x) \to (T\ \Box y)$

$\supset\text{-I}{:}\prod_{x,y:o}((T\ x) \to (T\ y)) \to (T\ (\supset xy))$

$\supset\text{-E}{:}\prod_{x,y:o}(T\ (\supset xy)) \to (T\ x) \to (T\ y)$

$\Box'\text{-I}{:}\prod_{x:o}\prod_{d:(T\ x)}(Cl\ x\ d) \to (T\ (\Box x))$

$Cl_{\Box'\text{-I}}{:}\prod_{x:o}\prod_{d_1:(T\ x)}\prod_{c_1:(Cl\ x\ d_1)}(Cl\ (\Box x)\ (\Box'\text{-I}\ x\ d_1\ c_1))$

$Cl_{\supset\text{-I}}{:}\prod_{x,y:o}\prod_{d:(T\ x)\to(T\ y)}\left(\prod_{a:(T\ x)}(Cl\ x\ a) \to (Cl\ y\ (da))\right) \to (Cl\ y\ (\supset\text{-I}\ x\ y\ d))$

$Cl_{\supset\text{-E}}{:}\prod_{x,y:o}\prod_{d_1:(T\ (\supset\ x\ y))}\prod_{d_2:(T\ x)}$
$(Cl\ x\ d_2) \to (Cl\ (\supset\ x\ y)\ d_1) \to (Cl\ y\ (\supset\text{-E}\ x\ y\ d_1\ d_2))$

$Cl_{\supset_{\Box}\text{-E}}{:}\prod_{x,y:o}\prod_{d_1:(T\ \Box(\supset xy))}\prod_{d_2:(T\ \Box x)}$
$(Cl\ \Box(\supset xy)\ d_1) \to (Cl\ \Box x\ d_2) \to (Cl\ \Box y\ (\supset_{\Box}\text{-E}\ x\ y\ d_1\ d_2))$

$\Box\text{-E}{:}\prod_{x:o}(T\ \Box x) \to (T\ x)$

$Cl_{\Box\text{-E}}{:}\prod_{x:o}\prod_{d:(T\ \Box x)}(Cl\ \Box\varphi\ d) \to (Cl\ x\ (\Box\text{-E}\ x\ d))$

$\ldots\ldots$

Figure 9: $\Sigma_{Cl}(\mathbf{NK})$ and its extensions for **NKT**, ...

5.2.3 "Boxed Assumptions"-judgement

In Figure 10 we give the signature $\Sigma_{\Box}(\mathbf{NS4})$, which adopts a special technique for implementing Prawitz' system **NS4** [27].

Given $\Delta \subseteq \Phi$ with $\mathrm{FV}(\Delta) \subseteq X$, we define the LF context $\gamma_{\Box}(\Delta)$ as follows:

$$\gamma_{\Box}(\Delta) \stackrel{\text{def}}{=} \begin{cases} \langle\rangle & \text{if } \Delta \equiv \emptyset \\ \gamma_{\Box}(\Delta'), v : (T\ \varepsilon_X(\varphi)) & \text{if } \Delta \equiv \Delta', \varphi,\ \varphi \text{ is not boxed and } v \text{ fresh for } \gamma_{\Box}(\Delta') \\ \gamma_{\Box}(\Delta'), v : (T\ \varepsilon_X(\varphi)), u : (Bx\ \varepsilon_X(\varphi)\ v) & \\ & \text{if } \Delta \equiv \Delta', \varphi,\ \varphi \text{ is boxed and } v, u \text{ are fresh for } \gamma_{\Box}(\Delta') \end{cases}$$

The long proof of adequacy relies upon some very technical lemmata. We report here only those needed for defining the encoding function; the others are in Section A.3.8. For sake of simplicity, we adopt the following definition: for p proof term and Γ context,

$$C(p,\Gamma) \stackrel{\text{def}}{=} \text{for all } v \in \mathrm{FV}(p),\ \text{if } (v{:}(T\ \varepsilon_X(\psi))) \in \Gamma \text{ then } \exists u.(u{:}(Bx\ \varepsilon_X(\psi)\ v)) \in \Gamma$$

Lemma 5.6 *Given a canonical term p s.t. $\Gamma_X, \gamma_{\Box}(\Delta) \vdash_{\Sigma_{\Box}(\mathbf{NS4})} p : (T\ t)$, if $C(p, \gamma_{\Box}(\Delta))$ holds then there is a canonical term b such that $\Gamma_X, \gamma_{\Box}(\Delta) \vdash_{\Sigma_{\Box}(\mathbf{NS4})} b : (Bx\ t\ p)$.*

A consequence of this lemma is the existence of a function $\beta_\Delta : \Lambda \to \Lambda$, where Δ is a LF context, inductively defined as follows:

$\beta_\Delta(v) \stackrel{\text{def}}{=} u$ if there exists t such that $(u{:}(Bx\ t\ v)) \in \Delta$

$\beta_\Delta(\supset\text{-I}\ t\ t'\ (\lambda v{:}(T\ t).p)) \stackrel{\text{def}}{=} (Bx_{\supset\text{-I}}\ t\ t'\ (\lambda v{:}(T\ t).p)(\lambda v{:}(T\ t)\lambda u{:}(Bx\ t\ v).\beta_{\Delta,(u:(Bx\ t\ v))}(p)))$

$\beta_\Delta(\supset\text{-E}\ t\ t'\ p_1\ p_2) \stackrel{\text{def}}{=} (Bx_{\supset\text{-E}}\ t\ t'\ p_1\ p_2\ \beta_\Delta(p_1)\ \beta_\Delta(p_2))$

$\beta_\Delta(\Box\text{-I}\ t\ p\ b) \stackrel{\text{def}}{=} (Bx_{\Box\text{-I}}\ t\ p\ b)$ $\qquad$ $\beta_\Delta(\Box\text{-E}\ t\ p) \stackrel{\text{def}}{=} (Bx_{\Box\text{-E}}\ t\ p\ \beta_\Delta(p))$

$\beta_\Delta(\supset_{\Box}\text{-I}\ t\ t'\ p') \stackrel{\text{def}}{=} (Bx_{\supset_{\Box}\text{-I}}\ t\ t'\ p'\ (\lambda v{:}(T\ t)\lambda u{:}(Bx\ t\ v).\beta_{\Delta,(u:(Bx\ t\ v))}(p)))$
where $p' \stackrel{\text{def}}{=} \lambda v{:}(T\ t)\lambda u{:}(Bx\ t\ v).p$

Judgements

$T{:}o \to Type$

$Bx{:}\prod_{x:o}(T\ x) \to Type$

Axioms and Rules

$\supset\text{-I}{:}\prod_{x,y:o}((T\ x) \to (T\ y)) \to (T(\supset\ x\ y))$

$\supset_{\Box}\text{-I}{:}\prod_{x,y:o}\left(\prod_{d:(T\ \Box x)}(Bx\ \Box x\ d) \to (T\ y)\right) \to (T\ (\supset\ \Box x\ y))$

$\supset\text{-E}{:}\prod_{x,y:o}(T\ (\supset\ x\ y)) \to (T\ x) \to (T\ y)$

$\Box\text{-I}{:}\prod_{x:o}\prod_{d:(T\ x)}(Bx\ x\ d) \to (T\ (\Box x))$

$\Box\text{-E}{:}\prod_{x:o}(T\ (\Box x)) \to (T\ x)$

$$Bx_{\supset_{\Box}\text{-I}}{:}\prod_{x,y:o}\ \prod_{d:(\prod_{a:(T\ \Box x)}(Bx\ \Box x\ a) \to (T\ y))}\left(\prod_{a:(T\ \Box x)}\ \prod_{b:(Bx\ \Box x\ a)}(Bx\ y\ (d\ a\ b))\right) \to (Bx\ (\supset \Box x\ y)\ (\supset_{\Box}\text{-I}\ x\ y\ d))$$

$$Bx_{\supset\text{-I}}{:}\prod_{x,y:o}\ \prod_{d:(T\ x) \to (T\ y)}\left(\prod_{a:(T\ x)}(Bx\ x\ a) \to (Bx\ y\ (da))\right) \to (Bx\ (\supset\ x\ y)\ (\supset\text{-I}\ x\ y\ d))$$

$$Bx_{\supset\text{-E}}{:}\prod_{x,y:o}\ \prod_{d_1:(T(\supset x\ y))}\ \prod_{d_2:(T\ x)}(Bx\ (\supset\ x\ y)\ d_1) \to (Bx\ x\ d_2) \to (Bx\ y\ (\supset\text{-E}\ x\ y\ d_1\ d_2))$$

$Bx_{\Box\text{-I}}{:}\prod_{x:o}\prod_{d:(T\ x)}\prod_{b:(Bx\ x\ d)}(Bx\ \Box x\ (\Box\text{-I}x\ d\ b))$

$Bx_{\Box\text{-E}}{:}\prod_{x:o}\prod_{d:(T\ \Box x)}(Bx\ \Box x\ d) \to (Bx\ x\ (\Box\text{-E}\ x\ d))$

Figure 10: $\Sigma_{\Box}(\mathbf{NS4})$.

This function maps each proof term p whose free variables are "boxed" (i.e., for each $v{:}(T\ t)$ free in p there is an assumption $u{:}(Bx\ t\ v)$ in Δ) in the corresponding proof term witnessing that p depends only on boxed assumptions. This function is well defined, because the v's of the pairs we introduce in Δ at the cases for $\supset$-I, $\supset_{\Box}$-I are "fresh".

Lemma 5.7 *$\forall X, \Delta, \varphi$, if $(X, \Delta) \models_{\mathbf{NS4}} \pi : \varphi$ then there exists a canonical form p such that $\Gamma_X, \gamma_{\Box}(\Delta) \vdash_{\Sigma_{\Box}(\mathbf{NS4})} p : (T\ \varepsilon_X(\varphi))$.*

A consequence of this lemma is the existence of the function $\varepsilon^{\Sigma_{\Box}(\mathbf{NS4})}_{X,\Delta}$, which maps proofs of **NS4** to canonical proof terms. This function is inductively defined as follows.

$$\varepsilon^{\Sigma_{\Box}(\mathbf{NS4})}_{X,\Delta}(\varphi) \stackrel{\text{def}}{=} v_{\varphi}$$

$$\varepsilon^{\Sigma_{\Box}(\mathbf{NS4})}_{X,\Delta}(\supset\text{-I}_{\varphi\psi}(\pi')) \stackrel{\text{def}}{=} \begin{cases} (\supset_{\Box}\text{-I}\ \varepsilon_X(\varphi)\ \varepsilon_X(\psi) \\ (\lambda v{:}(T\ \varepsilon_X(\varphi))\lambda u{:}(Bx\varepsilon_X(\varphi)v).\varepsilon^{\Sigma_{\Box}(\mathbf{NS4})}_{X,\Delta,\varphi}(\pi'))) & \text{if } \varphi \text{ boxed} \\ (\supset\text{-I}\ \varepsilon_X(\varphi)\ \varepsilon_X(\psi)\ (\lambda v{:}(T\varepsilon_X(\varphi)).\varepsilon^{\Sigma_{\Box}(\mathbf{NS4})}_{X,\Delta,\varphi}(\pi'))) & \text{if } \varphi \text{ not boxed} \end{cases}$$

$$\varepsilon^{\Sigma_{\Box}(\mathbf{NS4})}_{X,\Delta}(\supset\text{-E}_{\varphi,\psi}(\pi',\pi'')) \stackrel{\text{def}}{=} (\supset\text{-E}\ \varepsilon_X(\varphi)\ \varepsilon_X(\psi)\ \varepsilon^{\Sigma_{\Box}(\mathbf{NS4})}_{X,\Delta}(\pi'')\ \varepsilon^{\Sigma_{\Box}(\mathbf{NS4})}_{X,\Delta}(\pi'))$$

$$\varepsilon^{\Sigma_{\Box}(\mathbf{NS4})}_{X,\Delta}(\Box\text{-I}_{\varphi}(\pi')) \stackrel{\text{def}}{=} (\Box\text{-I}\ \varepsilon_X(\varphi)\ \varepsilon_{X,\Delta}(\pi')\ \beta_{\gamma_{\Box}(\Delta)}(\varepsilon^{\Sigma_{\Box}(\mathbf{NS4})}_{X,\Delta}(\pi')))$$

$$\varepsilon^{\Sigma_{\Box}(\mathbf{NS4})}_{X,\Delta}(\Box\text{-E}_{\varphi}(\pi')) \stackrel{\text{def}}{=} (\Box\text{-E}\ \varepsilon_X(\varphi)\ \varepsilon^{\Sigma_{\Box}(\mathbf{NS4})}_{X,\Delta}(\pi'))$$

Theorem 5.8 *The function $\varepsilon^{\Sigma_{\Box}(\mathbf{NS4})}_{X,\Delta}$ is a compositional bijection between proofs π, such that $(X, \Delta) \models_{\mathbf{NS4}} \pi : \varphi$, and canonical terms t, such that $\Gamma_X, \gamma_{\Box}(\Delta) \vdash_{\Sigma_{\Box}(\mathbf{NS4})} t : (T\ \varepsilon_X(\varphi))$.*

In this signature, besides a rule for establishing the "boxed assumption"-judgement corresponding to each rule in **NS4**, there is also an extra rule, namely $\supset_{\Box}$-I. This subtle rule is necessary in order to discharge "boxed assumption"-judgements: see the following example.

Judgements
$T{:}o \to \mathit{Type}$
$BF{:}\prod_{x:o}(T\ x) \to \mathit{Type}$
Axioms and Rules
$\supset\text{-I}{:}\prod_{x,y:o}((T\ x) \to (T\ y)) \to (T\ (\supset\ x\ y))$
$\supset_{\Box}\text{-I}{:}\prod_{x,y:o}\left(\prod_{d:(T\ \Box x)}(BF\ \Box x\ d) \to (T\ y)\right) \to (T\ (\supset\ \Box x\ y))$
$\supset\text{-E}{:}\prod_{x,y:o}(T\ (\supset\ x\ y)) \to (T\ x) \to (T\ y)$
$\Box\text{-I}{:}\prod_{x:o}\prod_{d:(T\ x)}(BF\ x\ d) \to (T\ (\Box x))$
$\Box\text{-E}{:}\prod_{x:o}(T\ \Box x) \to (T\ x)$
$BF_{\supset_{\Box}\text{-I}}{:}\prod_{x,y:o}\prod_{d:\prod_{a:(T\ \Box x)}(BF\ \Box x\ a) \to (T\ y)}\left(\prod_{a:(T\ \Box x)}\prod_{b:(BF\ \Box x\ a)}(BF\ y\ (d\ a\ b))\right) \to (BF\ (\supset\ \Box x\ y)\ (\supset_{\Box}\text{-I}\ x\ y\ d))$
$BF_{\supset\text{-I}}{:}\prod_{x,y:o}\prod_{d:(T\ x)\to(T\ y)}\left(\prod_{a:(T\ x)}(BF\ x\ a) \to (BF\ y\ (d\ a))\right) \to (BF\ (\supset\ x\ y)\ (\supset\text{-I}\ x\ y\ d))$
$BF_{\supset\text{-E}}{:}\prod_{x,y:o}\prod_{d_1:(T\ (\supset\ x\ y))}\prod_{d_2:(T\ x)}(BF\ (\supset\ x\ y)\ d_1) \to (BF\ x\ d_2) \to (BF\ y\ (\supset\text{-E}\ x\ y\ d_1\ d_2))$
$BF'_{\supset\text{-E}}{:}\prod_{x,y:o}\prod_{d_1:(T\ (\supset\ x\ \Box y))}\prod_{d_2:(T\ x)}(BF\ \Box y\ (\supset\text{-E}\ x\ \Box y\ d_1\ d_2))$
$BF_{\Box\text{-I}}{:}\prod_{x:o}\prod_{d:(T\ x)}\prod_{b:(BF\ x\ d)}(BF\ \Box x\ (\Box\text{-I}\ x\ d\ b))$
$BF_{\Box\text{-E}}{:}\prod_{x:o}\prod_{d:(T\ \Box x)}(BF\ x\ (\Box\text{-E}\ x\ d))$

Figure 11: $\Sigma_{Fr}(\mathbf{NS4})$.

Example 5.1 *We show the derivation of axiom* $4 : \Box\varphi \supset \Box\Box\varphi$*, both in* **NS4** *and in LF (for typographical reasons, we omit the function* ε_X*).*

$$\dfrac{\dfrac{\Box\varphi \vdash \Box\varphi}{\Box\varphi \vdash \Box\Box\varphi}\,\Box\text{-I}}{\vdash \Box\varphi \supset \Box\Box\varphi}\,\supset\text{-I}$$

$$\dfrac{\dfrac{\dfrac{\vdash \Gamma_X, v{:}(T\ \Box\varphi), u{:}(Bx\ \Box\varphi\ u)}{\Gamma_X, v{:}(T\ \Box\varphi), u{:}(Bx\ \Box\varphi\ u) \vdash (\Box\text{-I}\ \Box\varphi\ v\ u) : (T\ \Box\Box\varphi)}\,1}{\Gamma_X \vdash \lambda v{:}(T\ \Box\varphi)\lambda u{:}(Bx\ \Box\varphi\ v).(\Box\text{-I}\ \Box\varphi\ v\ u) : \prod_{v:(T\ \Box\varphi)}\prod_{u:(Bx\ \Box\varphi\ v)}(T\ \Box\Box\varphi)}\,2}{\Gamma_X \vdash (\supset_{\Box}\text{-I}\ \varphi\ \Box\Box\varphi\ \lambda v{:}(T\ \Box\varphi)\lambda u{:}(Bx\ \Box\varphi\ v).(\Box\text{-I}\ \Box\varphi\ v\ u)){:}(T\ (\Box\varphi \supset \Box\Box\varphi))}\,3$$

where $1 = app(\Box\text{-I})$*;* $2 = 2 \times abs$*;* $3 = app(\supset_{\Box}\text{-I})$*.*

5.2.4 "Boxed Fringe"-judgement

For the sake of completeness we sketch here how to encode Prawitz's *third version* of system **NS4** [27]. The signature $\Sigma_{Fr}(\mathbf{NS4})$ appears in Figure 11.

The judgement $BF : \prod_{x:o}(T\ x) \to \mathit{Type}$ holds only on proofs with a fringe of boxed formulæ (in the minimal fragment of modal logic, boxed formulæ are all the essentially modal formulæ). In the system there are rules for establishing the "boxed fringe" judgement corresponding to each rule in **NS4**. Additional rules for BF can be induced by elimination rules whenever the inferred formula is boxed (and hence belongs to the fringe). This is the case, e.g., of $\supset$-E.

$$\Sigma_{2j}(\mathbf{NK}')+ \begin{array}{|rcl|}\hline \multicolumn{3}{|l|}{\textbf{Judgements}}\\ Ta, V, T & : & o \to Type \\ \hline \multicolumn{3}{|l|}{\textbf{Axioms and Rules}}\\ C & : & \prod_{x:o}(V\ x) \to (T\ x)\\ \supset_T\text{-I} & : & \prod_{x,y:o}((T\ x) \to (T\ y)) \to (T\ (\supset xy))\\ \supset_T\text{-E} & : & \prod_{x,y:o}(T\ (\supset xy)) \to (T\ x) \to (T\ y)\\ & & \ldots\text{similarly for negation and } \mathit{ff}.\\ \hline \Box\text{-E} & : & \prod_{x:o}(Ta\ \Box x) \to (Ta\ x)\\ \Box_\Box\text{-I} & : & \prod_{x:o}(Ta\ \Box x) \to (Ta\ \Box\Box x)\\ \Box_\Diamond\text{-I} & : & \prod_{x:o}(Ta\ \Diamond x) \to (Ta\ \Box\Diamond x)\\ \Box_\supset\text{-E} & : & \prod_{x:o}(Ta\ \Box(\supset \Box(\supset x\ \Box x)\ x)) \to (Ta\ x)\\ \hline \end{array}$$

Figure 12: $\Sigma_{3j}(\mathbf{NK}'')$ and its extensions for $\mathbf{NK4}''$,...

5.2.5 Three-judgements

We can introduce ND-style systems for "truth" based on the multiple CR ND-style system $\mathbf{NK}'$ for validity. We need only to add a third consequence relation, namely $\Vdash$, with exactly the same rules as $\vdash$, and in addition the rule EMBED'. The whole system is called $\mathbf{NK}''$:

$$\mathbf{NK}'' \stackrel{\text{def}}{=} \mathbf{NK}' + \begin{array}{|cccc|}\hline \supset_T\text{-I}\ \dfrac{\Gamma, \varphi \Vdash \psi}{\Gamma \Vdash \varphi \supset \psi} & \supset_T\text{-E}\ \dfrac{\Gamma \Vdash \varphi \supset \psi \quad \Gamma \Vdash \varphi}{\Gamma \Vdash \psi} & & \text{RAA}_T\ \dfrac{\Gamma, \neg\varphi \Vdash \mathit{ff}}{\Gamma \Vdash \varphi} \\ \Gamma, \varphi \Vdash \varphi & \mathit{ff}_T\text{-I}\ \dfrac{\Gamma \Vdash \varphi \quad \Gamma \Vdash \neg\varphi}{\Gamma \Vdash \mathit{ff}} & \mathit{ff}_T\text{-E}\ \dfrac{\Gamma \Vdash \mathit{ff}}{\Gamma \Vdash \varphi} & \text{EMBED'}\ \dfrac{\Vdash \varphi}{\Vdash \varphi} \\ \hline \end{array}$$

Soundness of $\mathbf{NK}''$ is obvious; completeness follows from the fact that $\varphi_1, \ldots, \varphi_n \models \varphi$ iff $\varphi_1 \supset \ldots \supset \varphi_n \supset \varphi$ is valid.

In order to encode this system we add a judgement $T : o \to Type$, whose constructors are like those of Ta plus a constant C which represents the EMBED' rule (Figure 12). We can prove then

Theorem 5.9 *There is a compositional bijection between proofs* $\pi : \Delta \vdash_{\mathbf{NK}''} \varphi$ *with* $\mathrm{FV}(\pi) \subseteq X$ *and canonical terms* t *such that* $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{3j}(\mathbf{NK}'')} t : (T\ \varepsilon_X(\varphi))$.

Again, similarly to the case of two-judgement system for $\mathbf{K}$ (see Section 4.2.3), the resulting system is more powerful than this result points out, since it can deal with both truth and validity notions, at the same time. A suitable semantic counterpart of such a system is a *multiple consequence relation* (see [24, Chapter 3]), which combines those presented in Definition 1.1 with the "triple consequence relation" of Section 4.2.3. We introduce a pair of consequence relations as follows:

Definition 5.10 *Let* φ *range over formulæ and* Δ, Ξ *over sets of formulæ.*

- *the* multiple consequence relation *with respect to a model* $\mathcal{M}$ *is* $(\models^T_{\mathcal{M}}, \models^V_{\mathcal{M}})$ *where the two components are defined as follows:*

$$\Delta; \Xi \models^T_{\mathcal{M}} \varphi \iff (\forall s. s \models_{\mathcal{M}} \Delta) \Rightarrow (\forall s. s \models_{\mathcal{M}} \Xi \Rightarrow s \models_{\mathcal{M}} \varphi)$$
$$\Delta; \Xi \models^V_{\mathcal{M}} \varphi \iff (\forall s. s \models_{\mathcal{M}} \Delta) \Rightarrow (\forall s. s \models_{\mathcal{M}} \varphi)$$

- *the* (absolute) MCR *consists of the relations* $\models^T \stackrel{\text{def}}{=} \bigcap_{\mathcal{M}} \models^T_{\mathcal{M}}$, $\models^V \stackrel{\text{def}}{=} \bigcap_{\mathcal{M}} \models^V_{\mathcal{M}}$, *where* $\mathcal{M}$ *ranges over all modal models.*

This semantic consequence relation combines validity and truth CR's, faithfully to what is done by $\mathbf{NK}''$ at the syntactical level:

Theorem 5.11 *For $X \subset \Phi_a$, $\Delta, \Xi \subseteq \Phi_X$, $\varphi \in \Phi_X$, $J \in \{T, V\}$, the following are equivalent:*

1. *$\exists t$ canonical term such that $\Gamma_X, \gamma_{Ta}(\Delta), \gamma_T(\Xi) \vdash_{\Sigma_\Box(\mathbf{NK})} t : (J\ \varepsilon_X(\varphi))$;*
2. *$\Delta; \Xi \models^J \varphi$.*

The proof of this theorem follows the standard paradigm.

6 Cross soundness

As we have seen, different techniques can be used for encoding the same system; for instance, **K** can be encoded by using either "world parameters" ($\Sigma_w(\mathbf{K})$) or "no assumption"-judgements ($\Sigma_{Na}(\mathbf{K})$), or "two-judgments" ($\Sigma_{2j}(\mathbf{K})$). Morally, these techniques are closely related: for instance, (the encoding of) a proof has no assumptions (in $\Sigma_{Cl}(\mathbf{NK})$) iff it can be carried out from no assumptions (in $\Sigma_w(\mathbf{NK})$).

Theorem 6.1 (Cross-soundness for K) *For $X \subset \Phi_a$, $\Delta \subseteq \Phi_X$, $\varphi \in \Phi_X$, the following are equivalent:*

1. *$\exists t.\Gamma_X, \gamma_w(\emptyset) \vdash_{\Sigma_w(\mathbf{K})} t : (T\ w\ \varepsilon_X(\varphi))$*
2. *$\exists t', n.\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Na}(\mathbf{K})} n : (Na\ \varepsilon_X(\varphi)\ t')$*
3. *$\exists v.\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{2j}(\mathbf{K})} v : (V\ \varepsilon_X(\varphi))$*

Theorem 6.2 (Cross-soundness for NK) *For $X \subset \Phi_a$, $\Delta \subseteq \Phi_X$, $\varphi \in \Phi_X$, the following are equivalent:*

1. *$\exists t.\Gamma_X, \gamma_w(\emptyset) \vdash_{\Sigma_w(\mathbf{NK})} t : (T\ w\ \varepsilon_X(\varphi))$*
2. *$\exists t', n.\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Cl}(\mathbf{NK})} n : (Cl\ \varepsilon_X(\varphi)\ t')$*
3. *$\exists v.\Gamma_X, \gamma_{Ta}(\Delta) \vdash_{\Sigma_{3j}(\mathbf{NK})} v : (V\ \varepsilon_X(\varphi))$*

These results can be seen as "internal proofs" of adequacy of the encodings. Similar connections can be formulated with respect to other techniques appearing in this paper. These metatheoretic results could be proved formally within some Logical Framework, e.g. Coq.

7 Final Remarks

The work presented in this paper is essentially an anthology of encodings of Modal Logics in Logical Frameworks based on dependent-typed λ-calculus. In this sense it is part of a general project that we have carried out over the past ten years [15, 3, 17, 23, 24] aiming at exploring the expressive power of LF's. The implications and limitations of this approach to building proof assistants are not completely well-understood yet. As far as modal logics are concerned, many more practical experiments are called for. The various signatures presented in this paper should be compared between themselves and all of them should be compared with existing alternative implementations based on different philosophies.

However, we think that the practice of encoding logics in LF's has established at least one point. At the metalogical level, LF's are powerful tools for analyzing the compatibility of a given logic with the process of assuming and discharging hypothesis as well as the extent to which its rules can be presented schematically with respect to the constructors of the syntax. And taking into account these two aspects is essential in designing a nice ND-style system.

Finally, we feel that LF encodings can be naturally used also to teach logics, since LF provides a natural language for describing uniformly all the aspects of a logic, down to the tiniest detail.

Applications. Modalities are a common feature of most program logics [14, 16, 30], hence, the techniques we have presented here can be fruitfully employed in developing proof assistants for program logics. The "world parameter" technique was used for encoding a ND-style system for Dynamic Logic [17]. Applications of the other techniques presented in this papers deserve further investigations.

Comparison with Related Work. In recent years, several researches have addressed the problem of computer assisted proof search in the context of modal logics. For instance, Fitting, Simpson, Wallen have built new systems for modal logics which could be used as the basis of proof development environments, or even, theorem provers [10, 11, 29, 32]. Basin *et.al.*, Coen, Merz have developed packages for using Modal Logics and Temporal Logics within existing Logical Frameworks (namely, Isabelle) [5, 6, 22].

These approaches either utilize special formats or they are based on representations of Kripke semantics, or they do not address explicitly ND-style presentations. Truly sequent-like formats or tableaux formats are used in [6, 11, 32], while accessibility relations are exploited in [5, 29].

For instance, a thorough treatment of modal logics based on semantics is the one carried out by Basin and his co-authors in [5]. In this paper, Kripke semantics is built-in the calculus from the outset with great ingenuity: worlds are reified, and a first order proposition R over worlds is introduced in order to represent the accessibility relation. Introduction of modalities is then reduced to a quantification over accessible worlds; different axiomatizations of R are used to represent the various logics.

The work presented in this paper differs quite substantially from all these approaches. First of all we want to use a *standard general* Logical Framework so that we can re-use in our implementations pre-existing proof search tools, possibly developed for different systems. Moreover we work at a *purely syntactical* level. We try to achieve generality and independence from peculiar *ad hoc* mathematical constructions, which can be rather foreign to the proof developing experience.

Of course, it would be worthed exploring to what extent our rules could be derivable in semantically based frameworks. This would be especially interesting if the semantical apparatus, working in the background, could even be hidden from the user. Our feeling is that this is not so immediate in existing frameworks. This is the case of most proof rules; e.g., Prawitz's rule $\Box$-I for S4 (see Figure 2) is admissible but not derivable in a system such as the one of [5].

A Proofs

A.1 Proof of Theorems of Section 3

A.1.1 Proof of Theorem 3.1

The encoding function ε_X is clearly injective. It is easy to show by induction on the structure of formulæ that ε_X yields a canonical form of the appropriate type. Surjectivity is established by defining a decoding map δ_X that is left-inverse to ε_X. The decoding δ_X is defined by induction on the structure of the canonical forms as follows:

$$
\boxed{\begin{array}{rcl}
\delta_X(x) & \stackrel{\text{def}}{=} & x \quad \text{if } x \in X \\
\delta_X(\mathit{ff}) & \stackrel{\text{def}}{=} & \mathit{ff} \\
\delta_X(\neg\varphi) & \stackrel{\text{def}}{=} & \neg\delta_X(\varphi) \\
\delta_X(\Box\varphi) & \stackrel{\text{def}}{=} & \Box\delta_X(\varphi) \\
\delta_X(\supset \varphi\ \psi) & \stackrel{\text{def}}{=} & \delta_X(\varphi) \supset \delta_X(\psi)
\end{array}}
$$

Such δ_X is total, for [15, Lemma 2.4.4] and inspection of $\Sigma(\Phi)$ and Γ_X.

The compositionality property is established by a straightforward induction on the structure of modal formulæ (omitted). □

A.2 Proofs of Theorems of Section 4

A.2.1 Proof of Theorem 4.1

It is straightforward to verify by induction on the structure of proofs that, given the hypothesis of the theorem, $\varepsilon^{\Sigma(\mathbf{K}')}_{X,\Delta}(\pi)$ is a canonical term of type $(V\ \varepsilon_X(\varphi))$ in $\Sigma(\mathbf{K}')$ and $\Gamma_X, \gamma_V(\Delta)$. It is a routine matter to show by induction on proofs that $\varepsilon^{\Sigma(\mathbf{K}')}_{X,\Delta}$ is injective. To establish surjectivity we exhibit a left-inverse $\delta^{\Sigma(\mathbf{K}')}_{X,\Delta}$ defined by induction on the structure of the canonical forms as follows:

$$
\boxed{\begin{array}{l}
\delta^{\Sigma(\mathbf{K}')}_{X,\Delta}(v) \stackrel{\text{def}}{=} \delta_X(t), \qquad \mathit{if}\ (v{:}(V\ t)) \in \gamma_V(\Delta). \\
\delta^{\Sigma(\mathbf{K}')}_{X,\Delta}(A_1\ t'\ t'') \stackrel{\text{def}}{=} A_{1\,\delta_X(t'),\delta_X(t'')}, \qquad \text{Similarly for } A_2, A_3, K. \\
\delta^{\Sigma(\mathbf{K}')}_{X,\Delta}(\mathit{NEC}\ t\ p) \stackrel{\text{def}}{=} \mathrm{N{\scriptstyle EC}}_{\delta_X(t)}\left(\delta^{\Sigma(\mathbf{K}')}_{X,\Delta}(p)\right) \\
\delta^{\Sigma(\mathbf{K}')}_{X,\Delta}(\mathit{MP}\ t\ t'\ p\ p') \stackrel{\text{def}}{=} \mathrm{MP}_{\delta_X(t),\delta_X(t')}\left(\delta^{\Sigma(\mathbf{K}')}_{X,\Delta}(p), \delta^{\Sigma(\mathbf{K}')}_{X,\Delta}(p')\right)
\end{array}}
$$

This function is clearly total and well-defined. It remains to show that $\delta^{\Sigma(\mathbf{K}')}_{X,\Delta}\left(\varepsilon^{\Sigma(\mathbf{K}')}_{X,\Delta}(\pi)\right) = \pi$ and compositionality of the encoding; this is established by induction on the proofs. □

A.2.2 Proof of Theorem 4.2

We verify by induction on the structure of proofs that $\varepsilon^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(\pi)$ is a canonical term of type $(T\ w\ \varepsilon_X(\varphi))$ in $\Sigma_w(\mathbf{K})$ and $\Gamma_X, \gamma_w(\Delta)$.

Base Step. We have two cases. If π is instance of an axiom, say $\pi = A_{1\psi,\vartheta}$, then it is straightforward to prove that $\Gamma_X, \gamma_w(\Delta) \vdash_{\Sigma_w(\mathbf{K})} \varepsilon^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(A_{1\psi,\vartheta}){:}\,(T\ w\ \varepsilon_X(\psi \supset (\vartheta \supset \psi)))$. The cases of A_2, A_3, K are similar.

Otherwise, $\varphi \in \Delta$ is an assumption. Since $\varepsilon^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(\varphi) = v_\varphi \in \gamma_w(\Delta)$, immediately $\Gamma_X, \gamma_w(\Delta) \vdash_{\Sigma_w(\mathbf{K})} v_\varphi{:}(T\ w\ \varepsilon_X(\varphi))$.

Inductive Step. By cases on the last rule applied.

If $\pi \equiv \mathrm{MP}_{\psi,\varphi}(\pi', \pi'')$, then π', π'' are respectively valid proofs of $\psi \supset \varphi$, ψ w.r.t. (X, Δ). By inductive hypothesis, $\Gamma_X, \gamma_w(\Delta) \vdash_{\Sigma_w(\mathbf{K})} \varepsilon^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(\pi') : (T\ w\ \varepsilon_X(\psi))$ and
$\Gamma_X, \gamma_w(\Delta) \vdash_{\Sigma_w(\mathbf{K})} \varepsilon^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(\pi'') : (T\ w\ \varepsilon_X(\psi \supset \varphi))$. Therefore, we have immediately,

$$\Gamma_X, \gamma_w(\Delta) \vdash_{\Sigma_w(\mathbf{K})} \left(MP\ \varepsilon_X(\psi)\ \varepsilon_X(\varphi)\ w\ \varepsilon^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(\pi')\ \varepsilon^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(\pi'')\right) : (T\ w\ \varepsilon_X(\varphi)).$$

Otherwise, $\pi \equiv \textsc{Nec}_\varphi(\pi')$; then π' is a valid proof of φ w.r.t. $(X,\emptyset)$. By IH, we have that $\Gamma_X, \gamma_w(\emptyset) \vdash_{\Sigma_w(\mathbf{K})} \varepsilon^{\Sigma_w(\mathbf{K})}_{X,\emptyset,w}(\pi') : (T\ w\ \varepsilon_X(\varphi))$. By abstracting on w we have $\Gamma_X \vdash_{\Sigma_w(\mathbf{K})} \left(\lambda w'{:}U.\varepsilon^{\Sigma_w(\mathbf{K})}_{X,\emptyset,w'}(\pi')\right) : \prod_{w':U} (T\ w'\ \varepsilon_X(\varphi))$. Therefore, we have immediately

$$\Gamma_X, \gamma_w(\emptyset) \vdash_{\Sigma_w(\mathbf{K})} \left(\mathit{NEC}\ \varepsilon_X(\varphi)\ (\lambda w'{:}U.\varepsilon^{\Sigma_w(\mathbf{K})}_{X,\emptyset,w'}(\pi'))\ w\right) : (T\ w\ \Box\varepsilon_X(\varphi)).$$

By the above steps, it is easy to show that $\varepsilon^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}$ is injective. Surjectivity is established by exhibiting a left-inverse $\delta^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}$, defined by induction on the structure of the canonical forms as follows:

$\delta^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(v_\varphi) \stackrel{\text{def}}{=} \varphi,$	$\mathit{if}\ v_\varphi \in \mathit{dom}(\gamma_w(\Delta))$
$\delta^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(A_1\ t\ t'\ w) \stackrel{\text{def}}{=} A_{1\,\delta_X(t),\delta_X(t')},$	$\mathit{similarly\ for}\ A_2, A_3, K$
$\delta^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(MP\ t\ t'\ w\ p\ p') \stackrel{\text{def}}{=} \mathrm{MP}_{\delta_X(t),\delta_X(t')}\left(\delta^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(p), \delta^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(p')\right)$	
$\delta^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(\mathit{NEC}\ t\ (\lambda w'{:}U.p)\ w) \stackrel{\text{def}}{=} \textsc{Nec}_{\delta_X(t)}\left(\delta^{\Sigma_w(\mathbf{K})}_{X,\emptyset,w}(p)\right)$	

The decoding map $\delta^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}$ is total and well-defined by the definition of canonical forms and inspection of the signature $\Sigma_w(\mathbf{K})$. By the lemma of characterization, a canonical form p of type $(T\ w\ t)$ must have the shape $(\zeta M_1 \dots M_k)$, where k is the arity of ζ. By inspection of $\Sigma_w(\mathbf{K})$ and $\Gamma_X, \gamma_w(\Delta)$ we see that the only choices are $\zeta \in \{v_\varphi, A_1, A_2, A_3, K, MP, \mathit{NEC}\}$.
Base Step. We have two cases. If $p \equiv v_\varphi : (T\ w\ \varepsilon_X(\varphi))$ then, taken $\pi = \varphi$ we have a valid proof of φ w.r.t. (X,Δ). Otherwise, $p \in \{A_1, A_2, A_3, K\}$, say $t \equiv A_1\ t'\ t''\ w : (T\ w(\supset t'(\supset t''t')))$. Then we consider $\pi = A_{1\,\delta_X(t'),\delta_X(t'')}$. Similarly in the case p is A_2, A_3, K.
Inductive Step. We have two cases. If $p \equiv (MP\ t'\ t''\ w\ p'\ p'')$:$(T\ w\ t'')$, since p is well-typed, we have that $\Gamma_X, \gamma_w(\Delta) \vdash_{\Sigma_w(\mathbf{K})} p'{:}(T\ w\ t')$ and $\Gamma_X, \gamma_w(\Delta) \vdash_{\Sigma_w(\mathbf{K})} p''{:}(T\ w\ (\supset t'\ t''))$. By IH there are two proofs such that $(X,\Delta) \models_{\mathbf{K}} \delta^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(p'){:}\delta_X(t')$ and $(X,\Delta) \models_{\mathbf{K}} \delta^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(p''){:}\delta_X(\supset t'\ t'')$. Therefore by applying MP we obtain $(X,\Delta) \models_{\mathbf{K}} \pi{:}\delta_X(t'')$.
Otherwise, $p \equiv (\mathit{NEC}\ t'\ (\lambda w'{:}U.p')w) : (T\ w\ (\Box t'))$. Since p is well-typed, we have that $\Gamma_X, \gamma_w(\Delta) \vdash_{\Sigma_w(\mathbf{K})} (\lambda w'{:}U.p'){:}\prod_{w':U}(Tw't')$. Notice that each canonical term p of type $(T\ w\ t)$ has exactly one free variable of type U, namely w. This can be proved by induction on the structure of p (look at the previous steps). Hence, $(\lambda w'{:}U.p')$ has no free variable of type U. We can drop therefore the hypotheses $\gamma_w(\Delta)$, since if they appear free in p there should be two free variables of type U in p' — a contradiction. Hence, $\Gamma_X \vdash_{\Sigma_w(\mathbf{K})} (\lambda w'{:}U.p'){:}\prod_{w':U}(T\ w'\ t')$, that is $\Gamma_X, w'{:}U \vdash_{\Sigma_w(\mathbf{K})} p'{:}(T\ w'\ t')$. By IH there is a valid proof $(X,\emptyset) \models_{\mathbf{K}} \delta^{\Sigma_w(\mathbf{K})}_{X,\emptyset,w'}(p'){:}\delta_X(t')$. Hence by applying Nec we obtain $(X,\Delta) \models_{\Sigma_w(\mathbf{K})} \pi{:}\delta_X(\Box t')$.

It remains to show that $\delta^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}\left(\varepsilon^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}(\pi)\right) = \pi$, and that $\varepsilon^{\Sigma_w(\mathbf{K})}_{X,\Delta,w}$ is compositional. This is proved by induction on the structure of π, following the steps above. □

A.2.3 Proof of Lemma 4.3

By lemma of characterization, a canonical form p of type $(T\ t)$ must have the form $\zeta M_1 \dots M_k$, where k is the arity of ζ. By inspection of $\Sigma_{Na}(\mathbf{K})$ and Γ_X we see that the only choices for ζ are $\zeta \in \{A_1, A_2, A_3, K, MP, \mathit{NEC}\}$.
Base Step: p is an instance of an axiom scheme; say $p \equiv (A_1\ t\ t')$; we take $n = (Na_{A_1}\ t\ t')$. The cases of schemata A_2, A_3, K are similar.

Inductive Step. We have two cases.

If $p \equiv (MP\ t\ t'\ p'\ p'')$, since p is well-typed we have that $\Gamma_X \vdash_{\Sigma_{Na}(\mathbf{K})} p'{:}(T\ (\supset\ t\ t'))$ and $\Gamma_X \vdash_{\Sigma_{Na}(\mathbf{K})} p''{:}(T\ t)$. By IH there are n', n'' such that $\Gamma_X \vdash_{\Sigma_{Na}(\mathbf{K})} n'{:}(Na\ (\supset\ t\ t')\ p')$ and $\Gamma_X \vdash_{\Sigma_{Na}(\mathbf{K})} n''{:}(Na\ t\ p'')$. Then, $\Gamma_X \vdash_{\Sigma_{Na}(\mathbf{K})} (Na_{MP}\ t\ t'\ p''\ p'\ n''\ n') : (Na\ \ t\ (MP\ t\ t'\ p'\ p''))$.

Otherwise, $p \equiv (NEC\ t\ p'\ n)$; since p is well-typed we have that $\Gamma_X \vdash_{\Sigma_{Na}(\mathbf{K})} p'{:}(T\ t)$ and $\Gamma_X \vdash_{\Sigma_{Na}(\mathbf{K})} n{:}(Na\ t\ p')$. Then $\Gamma_X \vdash_{\Sigma_{Na}(\mathbf{K})} (Na_{NEC}\ t\ p'\ n){:}(Na\ \Box t\ (NEC\ t\ p'\ n))$. □

A.2.4 Proof of Theorem 4.4

It is straightforward to verify by induction on the structure of proofs that $\varepsilon^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}(\pi)$ is a canonical term of type $(T\ \varepsilon_X(\varphi))$ in $\Sigma_{Na}(\mathbf{K})$ and $\Gamma_X, \gamma_T(\Delta)$.

Base Step. We have two cases. If φ is an axiom instance, say $\pi \equiv A1_{\psi,\vartheta}$, then we take $p = \varepsilon^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}(A1_{\psi,\vartheta})$, it is straightforward to prove that $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Na}(\mathbf{K})} p{:}(T\ \varepsilon_X(\psi \supset (\vartheta \supset \psi)))$. Similarly in the cases A_2, A_3, K.

Otherwise, φ is an assumption, say $\pi = \varphi$ con $\varphi \in \Delta$; then we take $p = v_\varphi = \varepsilon^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}(\varphi)$. It is straightforward to prove that $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Na}(\mathbf{K})} p{:}(T\ \varepsilon_X(\varphi))$.

Inductive Step. By cases on the last rule applied.

If $\pi \equiv \mathrm{MP}_{\psi,\varphi}(\pi', \pi'')$, then π', π'' are respectively valid proofs of $\psi \supset \varphi, \psi$ w.r.t. (X, Δ). By IH there are two canonical terms such that $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Na}(\mathbf{K})} \varepsilon^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}(\pi') : (T\varepsilon_X(\psi \supset \varphi))$ and $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Na}(\mathbf{K})} \varepsilon^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}(\pi''){:}(Tw\varepsilon_X(\psi))$. Therefore, we have immediately, $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Na}(\mathbf{K})} MP\ \varepsilon_X(\psi)\ \varepsilon_X(\varphi)\ \varepsilon^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}(\pi'')\ \varepsilon^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}(\pi'){:}(T\ \varepsilon_X(\varphi))$.

Otherwise, $\pi \equiv \mathrm{NEC}_\varphi(\pi')$; then, we have that π' is a valid proof of φ w.r.t. $(X, \emptyset)$. So by IH, $\Gamma_X, \gamma_T(\emptyset) \vdash_{\Sigma_{Na}(\mathbf{K})} \varepsilon^{\Sigma_{Na}(\mathbf{K})}_{X,\emptyset}(\pi') : (T\ \varepsilon_X(\varphi))$. Now, by Lemma 4.3 we obtain that there exists a term n such that $\Gamma_X \vdash_{\Sigma_{Na}(\mathbf{K})} n : \left(Na\ \varepsilon_X(\varphi)\ \varepsilon^{\Sigma_{Na}(\mathbf{K})}_{X,\emptyset}(\pi')\right)$. Then we have $\Gamma_X, \gamma(\Delta) \vdash_{\Sigma_{Na}(\mathbf{K})} \left(NEC\ \varepsilon_X(\varphi)\ \varepsilon^{\Sigma_{Na}(\mathbf{K})}_{X,\emptyset}(\pi')\ n\right) : (T\ \Box\varepsilon_X(\varphi))$.

By above, $\varepsilon^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}$ is injective. Surjectivity is established by exhibiting a left-inverse $\delta^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}$, defined by induction on canonical forms as follows:

$$
\begin{array}{|ll|}
\hline
\delta^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}(v) \stackrel{\text{def}}{=} \delta_X(t) & \textit{if}\ (v{:}(T\ t)) \in \gamma_T(\Delta) \\
\delta^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}(A_1\ t'\ t'') \stackrel{\text{def}}{=} A1_{\delta_X(t'),\delta_X(t'')}, & \text{similarly for } A_2, A_3, K. \\
\delta^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}(NEC\ t\ p\ n) \stackrel{\text{def}}{=} \mathrm{NEC}_{\delta_X(t)}(\delta^{\Sigma_{Na}(\mathbf{K})}_{X,\emptyset}(\beta(n))) & \\
\delta^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}(MP\ t\ t'\ p\ p') \stackrel{\text{def}}{=} \mathrm{MP}_{\delta_X(t),\delta_X(t')}\left(\delta^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}(p), \delta^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}(p')\right) & \\
\hline
\end{array}
$$

where β is the inverse of α (which is defined after Lemma 4.3), and maps each canonical proof term n such that $\Gamma_X \vdash_{\Sigma_{Na}(\mathbf{K})} n : (Na\ p\ t)$, to the corresponding proof term p, such that $\Gamma_X \vdash_{\Sigma_{Na}(\mathbf{K})} p : (T\ t)$. The definition of β follows the steps of the technical Lemma A.1 (see below), which can be seen as the converse of Lemma 4.3. By induction one can prove that α and β are inverses.

The decoding map $\delta^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}$ is total and well-defined by the definition of canonical forms and inspection of the signature $\Sigma_{Na}(\mathbf{K})$. By lemma of characterization, a canonical form p of type $(T\ t)$ must have the form $\zeta M_1 \dots M_k$, where k is the arity of ζ. By inspection of $\Sigma_{Na}(\mathbf{K})$ and $\Gamma_X, \gamma_T(\Delta)$ we see that the only choices are $\zeta \in \{v_\varphi, A_1, A_2, A_3, K, MP, NEC\}$.

Base Step. We have two cases. If φ is an assumption, say $p \equiv v_\varphi{:}(T\ w\ \varepsilon_X(\varphi))$, then, taken $\pi = \varphi$ we have a valid proof of φ w.r.t. (X, Δ).

Otherwise, $p \in \{A_1, A_2, A_3, K\}$; say $p \equiv (A_1\ t'\ t''):(T(\supset t'(\supset t''t'))$. Then we take $\pi = A_{1\,\delta_X(t'),\delta_X(t'')}$. Similarly in the other cases.

Inductive Step. We have two cases.

If $p \equiv (MP\ t'\ t''\ p'\ p''):(T\ w\ t'')$, since p is well-typed, $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Na}(\mathbf{K})} p':(T\ (\supset t'\ t''))$ and $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Na}(\mathbf{K})} p'':(T\ t')$. By IH there are two proofs such that $(X, \Delta) \models_{\mathbf{K}} \delta^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}(p') : \delta_X(\supset t'\ t'')$ and $(X, \Delta) \models_{\mathbf{K}} \delta^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}(p'') : \delta_X(t')$. By applying MP we obtain $(X, \Delta) \models_{\mathbf{K}} \pi{:}\delta_X(t'')$.

Otherwise, $p \equiv (NEC\ t'\ p'\ n) : (T\ (\Box t'))$; then, since p is well-typed, $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Na}(\mathbf{K})} p':(T\ t')$ and $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Na}(\mathbf{K})} n:(Na\ t'\ p')$. By Lemma A.1 there is π' such that $(X, \emptyset) \models_{\mathbf{K}} \pi' : \delta_X(t')$. By applying NEC to π' we obtain $(X, \Delta) \models_{\mathbf{K}} \pi{:}\delta_X(\Box t')$.

It remains to show that $\delta^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}\left(\varepsilon^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}(\pi)\right) = \pi$, and that $\varepsilon^{\Sigma_{Na}(\mathbf{K})}_{X,\Delta}$ is compositional. This is proved by induction on the structure of proofs. □

Lemma A.1 *$\forall n$ canonical:* $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Na}(\mathbf{K})} n : (Na\ t\ p) \Rightarrow \exists \pi.(X, \emptyset) \models_{\mathbf{K}} \pi : \delta_X(t)$.

Proof. By lemma of characterization, a canonical form p of type $(Na\ t\ p)$ must have the form $\zeta M_1 \dots M_k$, where k is the arity of ζ. By inspection of $\Sigma_{Na}(\mathbf{K})$ and $\Gamma_X, \gamma_T(\Delta)$ we see that the only choices are $\zeta \in \{Na_{A_1}, Na_{A_2}, Na_{A_3}, Na_K, Na_{MP}, Na_{NEC}\}$.

Base Step: n is one of Na_{A_1}, Na_{A_2}, Na_K, say $n = (Na_{A_1}\ t\ t')$. Then, $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Na}(\mathbf{K})} n{:}(Na\ (\supset t\ (\supset t'\ t))\ (A_1\ t\ t'))$; hence we take $\pi = A_{1\,\delta_X(t),\delta_X(t)}$. The cases of other schemata are similar.

Inductive Step. We have two cases.

If $n \equiv (Na_{MP}\ t\ t'\ p\ p'\ n'\ n'')$, then since n is well-typed we have that $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Na}(\mathbf{K})} n' : (Na\ t\ p)$ and $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Na}(\mathbf{K})} n''{:}(Na\ (\supset\ t\ t')\ p')$. By IH $(X, \emptyset) \models_{\mathbf{K}} \pi'{:}\delta_X(t)$ and $(X, \emptyset) \models_{\mathbf{K}} \pi''{:}\delta_X(\supset\ t\ t')$. Then we take $\pi = \mathrm{MP}_{\delta_X(t),\delta_X(t')}(\pi', \pi'')$ with $(X, \emptyset) \models_{\mathbf{K}} \pi{:}\delta_X(t')$.

Otherwise, $n \equiv (Na_{NEC}\ t\ p\ n')$; since n is well-typed we have that $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Na}(\mathbf{K})} n'{:}(Na\ t\ p)$. By IH, $(X, \emptyset) \models_{\mathbf{K}} \pi'{:}\delta_X(t)$; then we take $\pi = \mathrm{NEC}_{\delta_X(t)}(\pi')$ with $(X, \emptyset) \models_{\mathbf{K}} \pi{:}\delta_X(\Box t)$. □

A.2.5 Proof of Theorem 4.6

Similar to that of Theorem 4.1. □

A.3 Proofs of Theorems of Section 5

A.3.1 Proof of Theorem 5.1

The proof follows the standard methodology of [15]. We exhibit the encoding function, and its inverse, for the $\vdash$ CR (the case of validity CR is similar). These functions are defined by induction on the proofs in $\mathbf{NK'}$ and on the terms of $\Sigma_{2j}(\mathbf{NK'})$ respectively.

$$
\begin{array}{|l|}
\hline
\alpha^{\Sigma_{2j}(\mathbf{NK'})}_{X,\Delta}(\varphi) \stackrel{\mathrm{def}}{=} v \qquad \text{s.t. } (v{:}(Ta\ \varepsilon_X(\varphi))) \in \gamma_{Ta}(\Delta) \\
\alpha^{\Sigma_{2j}(\mathbf{NK'})}_{X,\Delta}(\supset\text{-I}_{\varphi,\psi}(\pi)) \stackrel{\mathrm{def}}{=} \supset\text{-I}\ \varepsilon_X(\varphi)\ \varepsilon_X(\psi)\ \lambda v{:}(Ta\ \varepsilon_X(\varphi)).\alpha^{\Sigma_{2j}(\mathbf{NK'})}_{X,(\Delta,\varphi)}(\pi) \\
\alpha^{\Sigma_{2j}(\mathbf{NK'})}_{X,\Delta}(\supset\text{-E}_{\varphi,\psi}(\pi', \pi'')) \stackrel{\mathrm{def}}{=} \supset\text{-E}_{Ta,Ta}\ \varepsilon_X(\varphi)\ \varepsilon_X(\psi)\ \alpha^{\Sigma_{2j}(\mathbf{NK'})}_{X,\Delta}(\pi')\ \alpha^{\Sigma_{2j}(\mathbf{NK'})}_{X,\Delta}(\pi'') \\
\alpha^{\Sigma_{2j}(\mathbf{NK'})}_{X,\Delta}(\supset_\Box\text{-E}_{\varphi,\psi}(\pi', \pi'')) \stackrel{\mathrm{def}}{=} \supset_\Box\text{-E}\ \varepsilon_X(\varphi)\ \varepsilon_X(\psi)\ \alpha^{\Sigma_{2j}(\mathbf{NK'})}_{X,\Delta}(\pi')\ \alpha^{\Sigma_{2j}(\mathbf{NK'})}_{X,\Delta}(\pi'') \\
\hline
\end{array}
$$

$$
\begin{array}{|l|}
\hline
\beta^{\Sigma_{2j}(\mathbf{NK'})}_{X,\Delta;\Psi}(v) \stackrel{\mathrm{def}}{=} \delta_X(t) \qquad \text{such that } (v{:}Ta\ t) \in \gamma_{Ta}(\Delta) \cup \Psi \\
\beta^{\Sigma_{2j}(\mathbf{NK'})}_{X,\Delta;\Psi}(\supset\text{-I}\ t\ t'\ \lambda v{:}(Ta\ t).p) \stackrel{\mathrm{def}}{=} \supset\text{-I}_{\delta_X(t),\delta_X(t')}(\beta^{\Sigma_{2j}(\mathbf{NK'})}_{X,\Delta;\Psi,(v,t)}(p)) \\
\beta^{\Sigma_{2j}(\mathbf{NK'})}_{X,\Delta,\Psi}(\supset\text{-E}_{Ta,Ta}\ t\ t'\ p\ p') \stackrel{\mathrm{def}}{=} \supset\text{-E}_{\delta_X(t),\delta_X(t')}(\beta^{\Sigma_{2j}(\mathbf{NK'})}_{X,\Delta,\Psi}(p), \beta^{\Sigma_{2j}(\mathbf{NK'})}_{X,\Delta,\Psi}(p')) \\
\beta^{\Sigma_{2j}(\mathbf{NK'})}_{X,\Delta,\Psi}(\supset_\Box\text{-E}\ t\ t'\ p\ p') \stackrel{\mathrm{def}}{=} \supset_\Box\text{-E}_{\delta_X(t),\delta_X(t')}(\beta^{\Sigma_{2j}(\mathbf{NK'})}_{X,\Delta,\Psi}(p), \beta^{\Sigma_{2j}(\mathbf{NK'})}_{X,\Delta,\Psi}(p')) \\
\hline
\end{array}
$$

A.3.2 Proof of Theorem 5.2

Very similar to Theorem 4.2. We have only to take care of the $\supset$-I rule, which involves a discharge, as a new case of inductive steps.

If $\pi \equiv \supset\text{-I}_{\varphi,\psi}(\pi')$, then $(X, (\Delta, \varphi)) \models_{\mathbf{NK}} \pi':\psi$. By IH, we have $\Gamma_X, \gamma_w(\Delta, \varphi) \vdash_{\Sigma_w(\mathbf{NK})} \varepsilon^{\Sigma_w(\mathbf{NK})}_{X,(\Delta,\varphi),w}(\pi') : (T\ w\ \varepsilon_X(\psi))$. By abstracting on v_φ, we obtain

$$\Gamma_X, \gamma_w(\Delta) \vdash_{\Sigma_w(\mathbf{NK})} (\lambda v{:}(T\ w\ \varepsilon_X(\varphi)).\varepsilon^{\Sigma_w(\mathbf{NK})}_{X,(\Delta,\varphi),w}(\pi')) : \prod_{v:(T\ w\ \varepsilon_X(\varphi))} (T\ w\ \varepsilon_X(\psi)).$$

By applying the constant $\supset$-I, we obtain

$$\Gamma_X, \gamma_w(\Delta) \vdash_{\Sigma_w(\mathbf{NK})} \supset\text{-I}\ \varepsilon_X(\varphi)\ \varepsilon_X(\psi)\ w\ (\lambda v{:}(T\ w\ \varepsilon_X(\varphi)).\varepsilon^{\Sigma_w(\mathbf{NK})}_{X,(\Delta,\varphi),w}(\pi')) : (T\ w\ \varepsilon_X(\supset\ \varphi\ \psi)).$$

The rest of the proof follows closely that of Theorem 4.2. We show just the left-inverse:

$$\begin{array}{|l|}\hline
\delta^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w}(\varphi) \stackrel{\text{def}}{=} \delta^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w;\emptyset}(\varphi) \\
\delta^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w;\Psi}(v) \stackrel{\text{def}}{=} \delta_X(t) \qquad\qquad , if\ (v{:}(T\ w\ t)) \in \gamma_w(\Delta) \cup \Psi \\
\delta^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w;\Psi}(\supset\text{-I}\ t\ t'\ w\ (\lambda v{:}(T\ w\ t).p')) \stackrel{\text{def}}{=} \supset\text{-I}_{\delta_X(t),\delta_X(t')}\left(\delta^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w;\Psi,(v:(T\ w\ t))}(p')\right) \\
\delta^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w;\Psi}(\supset_\Box\text{-E}\ t\ t'\ w\ p\ p') \stackrel{\text{def}}{=} \supset_\Box\text{-E}_{\delta_X(t),\delta_X(t')}\left(\delta^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w;\Psi}(p), \delta^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w;\Psi}(p')\right) \\
\delta^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w;\Psi}(\supset\text{-E}\ t\ t'\ w\ p\ p') \stackrel{\text{def}}{=} \supset\text{-E}_{\delta_X(t),\delta_X(t')}\left(\delta^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w;\Psi}(p), \delta^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w;\Psi}(p')\right) \\
\delta^{\Sigma_w(\mathbf{NK})}_{X,\Delta,w;\Psi}(\Box'\text{-I}\ t\ (\lambda w'{:}U.p)\ w) \stackrel{\text{def}}{=} \Box'\text{-I}_{\delta_X(t)}\left(\delta^{\Sigma_w(\mathbf{NK})}_{X,\emptyset,w';\emptyset}(p)\right) \\ \hline
\end{array} \Box$$

A.3.3 Proof of Lemma 5.3

By lemma of characterization, a canonical form p of type $(T\ t)$ must be $\zeta M_1 \ldots M_k$, where k is the arity of ζ. By inspection of $\Sigma_{Cl}(\mathbf{NK})$ and $\Gamma_X, \gamma_T(\Delta)$, we see that the only choices for ζ are $\zeta \in \{v_\varphi, \supset\text{-I}, \supset\text{-E}, \Box'\text{-I}, \supset_\Box\text{-E} \ldots\}$.

Base Step. If p is an assumption of type $(p{:}(T\ t)) \in \Delta$ then we have $p \in \text{FV}(p)$ and hence $c{:}(Cl\ t\ p) \in \Xi_p(\Delta)$.

Inductive Step. By cases on the last rule applied. We will see only some significant cases, the other being similar.

$\bullet p \equiv (\supset\text{-E}\ t\ t'\ p'\ p'')$: since p is well-typed we have that $\Gamma_X, \Delta \vdash_{\Sigma_{Cl}(\mathbf{NK})} p'{:}(T\ (\supset\ t\ t'))$ and $\Gamma_X, \Delta \vdash_{\Sigma_{Cl}(\mathbf{NK})} p''{:}(T\ t)$. By IH, $\Gamma_X, \Delta, \Xi_{p'}(\Delta) \vdash_{\Sigma_{Cl}(\mathbf{NK})} c'{:}(Cl\ (\supset\ t\ t')\ p')$ and $\Gamma_X, \Delta, \Xi_{p''}(\Delta) \vdash_{\Sigma_{Cl}(\mathbf{NK})} c''{:}(Cl\ t\ p'')$. Since $\Xi_p(\Delta) \supseteq \Xi_{p'}(\Delta), \Xi_{p''}(\Delta)$, then $\Gamma_X, \Delta, \Xi_p(\Delta) \vdash_{\Sigma_{Cl}(\mathbf{NK})} (Cl_{\supset\text{-E}}\ t\ t'\ p'\ p''\ c''\ c') : (Cl\ t'\ (\supset\text{-E}t\ t'\ p'\ p''))$.

$\bullet p \equiv (\supset\text{-I}\ t\ t'\ p')$: since p is well-typed we have $\Gamma_X, \Delta \vdash_{\Sigma_{Cl}(\mathbf{NK})} p'{:}(T\ t) \to (T\ t')$. Since p' is a canonical form, it must be $p' = \lambda x : (T\ t).p''$, where $\Gamma_X, \Delta, x{:}(T\ t) \vdash_{\Sigma_{Cl}(\mathbf{NK})} p'' : (T\ t')$. By IH, $\Gamma_X, \Delta, x{:}(T\ t), \Xi_{p''}(\Delta, x{:}(T\ t)) \vdash_{\Sigma_{Cl}(\mathbf{NK})} c''{:}(Cl\ t'\ p'')$. Now we have that $\Xi_{p''}(\Delta, x{:}(T\ t)) \subseteq \Xi_{p''}(\Delta), c'{:}(Cl\ t\ x)$, then by abstracting on c' and x we obtain

$$\Gamma_X, \Delta, \Xi_{p''}(\Delta) \vdash_{\Sigma_{Cl}(\mathbf{NK})} (\lambda x{:}(T\ t).\lambda c'{:}(Cl\ t\ x).c'') : \prod_{x:(T\ t)} (Cl\ t\ x) \to (Cl\ t'\ p'').$$

Moreover we have that $\Xi_p(\Delta) = \Xi_{p''}(\Delta)$ because $\text{FV}(p) = \text{FV}(p'')\setminus\{x\}$ and $x{:}(T\ t) \notin \Delta$ (otherwise $\Delta, x{:}(T\ t)$ would be not a valid context). Then, defining $t_1 \stackrel{\text{def}}{=} (\lambda x{:}(T\ t)\lambda c'{:}(Cl\ t\ x).c'')$, we have $\Gamma_X, \Delta, \Xi_p(\Delta) \vdash_{\Sigma_{Cl}(\mathbf{NK})} t_1 : \prod_{x:(T\ t)}(Cl\ t\ x) \to (Cl\ t'\ p'')$. We apply now $Cl_{\supset\text{-I}}$ obtaining $\Gamma_X, \Delta, \Xi_p(\Delta) \vdash_{\Sigma_{Cl}(\mathbf{NK})} (Cl_{\supset\text{-I}}\ t\ t'\ p'\ t_1) : (Cl\ t'\ (\supset\text{-I}\ t\ t'\ p'))$.

•$p \equiv (\Box'\text{-I}\ t\ p'\ c')$: since p is well-typed we have $\Gamma_X, \Delta \vdash_{\Sigma_{Cl}(\mathbf{NK})} p' : (T\ t)$ and $\Gamma_X, \Delta \vdash_{\Sigma_{Cl}(\mathbf{NK})} c' : (Cl\ t\ p')$. Then we apply $\Box'$-I obtaining

$$\Gamma_X, \Delta \vdash_{\Sigma_{Cl}(\mathbf{NK})} (Cl_{\Box'\text{-}\mathbf{I}}\ t\ p'\ c') : (Cl\ (\Box t)(\Box'\text{-I}\ t\ p'\ c'))$$

and therefore $\Gamma_X, \Delta, \Xi_p(\Delta) \vdash_{\Sigma_{Cl}(\mathbf{NK})} (Cl_{\Box'\text{-}\mathbf{I}}\ t\ p'\ c') : (Cl\ (\Box t)\ (\Box'\text{-I}\ t\ p'\ c'))$. □

A.3.4 Proof of Lemma 5.4

By lemma of characterization, c of type $(Cl\ t\ p)$ is of the form $\zeta M_1 \ldots M_k$, where k is the arity of ζ. By inspection of $\Sigma_{Cl}(\mathbf{NK})$ and $\Gamma_X, \gamma_T(\Delta)$ we see that the only choices are $\zeta \in dom(\Xi) \cup \{Cl_{\supset\text{-I}}, Cl_{\supset\text{-E}}, Cl_{\Box'\text{-I}}, Cl_{\supset_\Box\text{-E}} \cdots\}$.

Base Step: $c{:}(Cl\ t'\ p)) \in \Xi$; then, the claim is trivial.

Inductive Step: by cases on the top constructor. We see only some significant cases, the other being similar.

•$c \equiv (Cl_{\supset\text{-}\mathbf{I}}\ t\ t'\ p\ t'') : (Cl\ t'\ (\supset\text{-I}\ t\ t'\ p))$: since c is well-typed we have that $\Gamma_X, \Delta, \Xi \vdash_{\Sigma_{Cl}(\mathbf{NK})} p{:}(T\ t) \to (T\ t')$ and $\Gamma_X, \Delta, \Xi \vdash_{\Sigma_{Cl}(\mathbf{NK})} t'' : \prod_{x:t}(Cl\ t\ x) \to (Cl\ t'\ (p\ x))$. Since t'' is a canonical form then it must be $t'' = \lambda x{:}(T\ t)\lambda c'{:}(Cl\ t\ x).t'''$. Then by some introductions we obtain $\Gamma_X, \Delta, x{:}(T\ t), \Xi, c'{:}(Cl\ t\ x) \vdash_{\Sigma_{Cl}(\mathbf{NK})} t'''{:}(Cl\ t'\ (p\ x))$. By the IH we know $\Gamma_X, \Delta'', \Xi, c'{:}(Cl\ t\ x) \vdash_{\Sigma_{Cl}(\mathbf{NK})} t'''{:}(Cl\ t'\ (p\ x))$, where $\Delta'' \stackrel{\text{def}}{=} \{p{:}(T\ t') | (Cl\ t'\ p) \in \Im(\Xi, c'{:}(Cl\ t\ x))\} = \Delta' \cup \{x{:}(T\ t)\}$. Then, by abstracting on x, c' we find that $\Gamma_X, \Delta', \Xi \vdash_{\Sigma_{Cl}(\mathbf{NK})} t'' : \prod_{x:(T\ t)}(Cl\ t\ x) \to (Cl\ t'\ (p\ x))$. Finally, by applying $Cl_{\supset\text{-}\mathbf{I}}$ we obtain $\Gamma_X, \Delta', \Xi \vdash_{\Sigma_{Cl}(\mathbf{NK})} (Cl_{\supset\text{-}\mathbf{I}}\ t\ t'\ p\ t''){:}(Cl\ t'\ (\supset\text{-I}\ t\ t'\ p))$.

•$c \equiv (Cl_{\Box'\text{-}\mathbf{I}}\ t\ p\ c') : (Cl\ \Box t\ (\Box'\text{-I}\ t\ p\ c'))$: since c is well-typed we have that $\Gamma_X, \Delta, \Xi \vdash_{\Sigma_{Cl}(\mathbf{NK})} p{:}(T\ t)$ and $\Gamma_X, \Delta, \Xi \vdash_{\Sigma_{Cl}(\mathbf{NK})} c'{:}(Cl\ t\ p)$. By IH we have that $\Gamma_X, \Delta', \Xi \vdash_{\Sigma_{Cl}(\mathbf{NK})} c'{:}(Cl\ t\ p)$. Hence, $\Gamma_X, \Delta', \Xi \vdash_{\Sigma_{Cl}(\mathbf{NK})} p{:}(T\ t)$. Then we apply $Cl_{\Box'\text{-}\mathbf{I}}$ obtaining $\Gamma_X, \Delta', \Xi \vdash_{\Sigma_{Cl}(\mathbf{NK})} (Cl_{\Box'\text{-}\mathbf{I}}\ t\ p\ c') : (Cl\ \Box t\ (\Box'\text{-I}\ t\ p\ c'))$.

•$c \equiv (Cl_{\supset_\Box\text{-E}}\ t\ t'\ p\ p'\ c'\ c'')$: an immediate application of IH on c', c''. □

A.3.5 Proof of Theorem 5.5

It is straightforward to verify by induction on the structure of proofs that, given the hypothesis of the theorem, $\varepsilon^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta}(\pi)$ is a canonical term of type $(T\ \varepsilon_X(\varphi))$ in $\Sigma_{Cl}(\mathbf{NK})$ and $\Gamma_X, \gamma_T(\Delta)$.

Base Step: φ is an assumption, i.e. $\pi = \varphi \in \Delta$. Then immediately $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Cl}(\mathbf{NK})} v_\varphi{:}(T\ \varepsilon_X(\varphi))$.

Inductive Step. By cases on the last rule applied. We see only some significant cases, the other being similar.

•$\pi \equiv \supset\text{-I}_{\varphi,\psi}(\pi')$: then $(X, (\Delta, \varphi)) \models_{\mathbf{NK}} \pi'{:}\psi$. By IH we have that $\Gamma_X, \gamma_T(\Delta, \varphi) \vdash_{\Sigma_{Cl}(\mathbf{NK})} t'{:}(T\ \varepsilon_X(\psi))$. Let $t'' \stackrel{\text{def}}{=} \lambda v_\varphi{:}(T\ \varepsilon_X(\varphi)).t'$; then $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Cl}(\mathbf{NK})} t'' : \prod_{v_\varphi:(T\ \varepsilon_X(\varphi))}(T\ \varepsilon_X(\psi))$. By applying $\supset$-I we obtain $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Cl}(\mathbf{NK})} (\supset\text{-I}\ \varepsilon_X(\varphi)\ \varepsilon_X(\psi)\ t''){:}(T\ (\supset\ \varepsilon_X(\varphi)\ \varepsilon_X(\psi)))$.

•$\pi \equiv \supset\text{-E}_{\varphi,\psi}(\pi', \pi'')$: then $(X, \Delta) \models_{\mathbf{NK}} \pi' : \varphi \supset \psi$ and $(X, \Delta) \models_{\mathbf{NK}} \pi'' : \varphi$. By IH, $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Cl}(\mathbf{NK})} t' : (T\ \varepsilon_X(\varphi \supset \psi))$ and $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Cl}(\mathbf{NK})} t'' : (T\ \varepsilon_X(\varphi))$. Therefore by applying $\supset$-E we obtain $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Cl}(\mathbf{NK})} (\supset\text{-E}\ \varepsilon_X(\varphi)\ \varepsilon_X(\psi)\ t'\ t''){:}(T\ \varepsilon_X(\psi))$.

•$\pi \equiv \Box'\text{-I}_\varphi(\pi')$: then $(X, \emptyset) \models_{\mathbf{NK}} \pi' : \varphi$. By IH, $\Gamma_X \vdash_{\Sigma_{Cl}(\mathbf{NK})} t_1 : (T\ \varepsilon_X(\varphi))$, and hence by Lemma 5.3 there is a term c_{t_1} such that $\Gamma_X \vdash_{\Sigma_{Cl}(\mathbf{NK})} c_{t_1}{:}(Cl\ \varepsilon_X(\varphi)\ t_1)$. Therefore, by applying $\Box'$-I, we obtain $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Cl}(\mathbf{NK})} (\Box'\text{-I}\ \varepsilon_X(\varphi)\ t_1\ c_{t_1}) : (T\ \Box\varepsilon_X(\varphi))$.

By the above steps, it is easy to see that $\varepsilon^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta}$ is injective. Surjectivity is established by exhibiting a left-inverse $\delta^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta}$, defined by induction on the structure of the canonical forms as follows:

$$\begin{array}{|ll|}\hline
\delta^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta}(\varphi) \stackrel{\text{def}}{=} \delta^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta;\emptyset}(\varphi) & \\
\delta^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta;\Psi}(v) \stackrel{\text{def}}{=} \delta_X(t) & ,if\ (v{:}(T\ t)) \in \gamma_T(\Delta) \cup \Psi \\
\delta^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta;\Psi}(\Box'\text{-I}\ t\ p\ c) \stackrel{\text{def}}{=} \Box'\text{-I}_{\delta_X(t)}(\delta^{\Sigma_{Cl}(\mathbf{NK})}_{X,\emptyset;\emptyset}(p)) & \\
\delta^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta;\Psi}(\supset\text{-I}\ t\ t'\ (\lambda v{:}(T\ t).p)) \stackrel{\text{def}}{=} \supset\text{-I}_{\delta_X(t),\delta_X(t')}(\delta^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta;\Psi,(v:(T\ t))}(p)) & \\
\delta^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta;\Psi}(\supset\text{-E}\ t\ t'\ p\ p') \stackrel{\text{def}}{=} \supset\text{-E}_{\delta_X(t),\delta_X(t')}(\delta^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta;\Psi}(p), \delta^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta;\Psi}(p')) & \\
\delta^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta;\Psi}(\supset_\Box\text{-E}\ t\ t'\ p\ p') \stackrel{\text{def}}{=} \supset_\Box\text{-E}_{\delta_X(t),\delta_X(t')}(\delta^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta;\Psi}(p), \delta^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta;\Psi}(p')) & \\ \hline
\end{array}$$

The map $\delta^{\Sigma_{Cl}(\mathbf{NK})}_{X,\Delta}$ is total and well-defined for the definition of canonical forms and inspection of the signature $\Sigma_{Cl}(\mathbf{NK})$. The application of $\Box'$-I is sound, for the presence of $c : (Cl\ \delta_X(t)\ p)$ and the fact that no Cl assumptions are made by the encoding of the context $(\gamma_T(\Delta))$.

By lemma of characterization, a canonical form p of type $(T\ t)$ must be $\zeta M_1 \dots M_k$, where k is the arity of ζ. By inspection of $\Sigma_{Cl}(\mathbf{NK}), \Gamma_X, \gamma_T(\Delta)$ we see that the only choices for ζ are $\zeta \in \{v_\varphi, \Box'\text{-I}, \supset\text{-I}, \supset\text{-E}, \supset_\Box\text{-E} \dots\}$.
Base Step: $p = v_\varphi \in \gamma_T(\Delta)$, then we take $\pi = \varphi$.
Inductive Step: we see only some significant cases.

•$p \equiv \Box'\text{-I}\ t\ p'\ c$: since p is well-typed we have that $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Cl}(\mathbf{NK})} p' : (T\ t)$ and $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Cl}(\mathbf{NK})} c : (Cl\ t\ p')$. By Lemma 5.4, there is a term c such that $\Gamma_X, \emptyset \vdash_{\Sigma_{Cl}(\mathbf{NK})} c : (Cl\ t\ p')$; since c is well-typed, $\Gamma_X, \emptyset \vdash_{\Sigma_{Cl}(\mathbf{NK})} p'{:}(T\ t)$. By the IH, we obtain that there exists π' such that $(X, \emptyset) \models_{\mathbf{NK}} \pi' : \delta_X(t)$ and hence we conclude $\pi = \Box'\text{-I}_{\delta_X(t)}(\pi')$.

•$p \equiv \supset\text{-I}\ t\ t'\ p'$: since p is well-typed, $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Cl}(\mathbf{NK})} p' : (T\ t) \to (T\ t')$, then $\Gamma_X, \gamma_T(\Delta), a : (T\ t) \vdash_{\Sigma_{Cl}(\mathbf{NK})} p'a : (T\ t')$. By IH there exists π' such that $(X, (\Delta, \delta_X(t))) \models_{\mathbf{NK}} \pi'{:}\delta_X(t')$, and hence we conclude taking $\pi = \supset\text{-I}_{\delta_X(t),\delta_X(t')}(\pi')$.

•$p \equiv \supset\text{-E}\ t\ t'\ p'\ p''$: since p is well-typed, $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Cl}(\mathbf{NK})} p' : (T\ (\supset\ t\ t'))$ and $\Gamma_X, \gamma_T(\Delta) \vdash_{\Sigma_{Cl}(\mathbf{NK})} p''{:}(T\ t)$. By IH there exist π', π'' such that $(X, \Delta) \models_{\mathbf{NK}} \pi' : \delta_X(\supset\ t\ t')$ and $(X, \Delta) \models_{\mathbf{NK}} \pi'' : \delta_X(t)$. We conclude taking $\pi = \supset\text{-E}_{\delta_X(t),\delta_X(t')}(\pi', \pi'')$. □

A.3.6 Proof of Lemma 5.6

By the lemma of characterization, a canonical form p of type $(T\ t)$ is $\zeta M_1 \dots M_k$, where k is the arity of ζ, which is $\zeta \in \{v_\varphi, \supset\text{-I}, \supset\text{-E}, \Box\text{-I}, \Box\text{-E}, \supset_\Box\text{-I} \dots\}$.
Base Step: $p \equiv v : (T\ t)$. By definition of $\gamma_\Box$, $C(p, \gamma_\Box(\Delta))$ holds, then there is the assumption $(u{:}(Bx\ t\ v_{\delta_X(t)})) \in \gamma_\Box(\Delta)$. Hence, $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} u : (Bx\ t\ v)$.
Inductive Step: we see only some significant cases.

•$p \equiv (\Box\text{-I}\ t\ p_1\ b) : (T\ \Box t)$. Since p is well-typed, we have $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} p_1 : (T\ t)$ and $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} b : (Bx\ t\ p_1)$. Hence $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} (Bx_{\Box\text{-}\mathbf{I}}\ t\ p_1\ b) : (Bx\ \Box t\ (\Box\text{-I}\ t\ p_1\ b))$.

•$p \equiv (\supset\text{-I}\ t\ t'\ (\lambda v{:}(T\ t).p_1)) : (T\ (\supset\ t\ t'))$: since p is well-typed, $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} (\lambda v{:}(T\ t).p_1) : (\Pi_{v:(T\ t)}(T\ t'))$, that is $\Gamma_X, \gamma_\Box(\Delta), v : (T\ t) \vdash_{\Sigma_\Box(\mathbf{NS4})} p_1 : (T\ t')$. Moreover, chosen a fresh variable u, $C\,(p_1, (\gamma(\Delta), v : (T\ t), u : (Bx\ t\ v)))$ holds. Then, by IH there is b_1 such that $\Gamma_X, \gamma_\Box(\Delta), v{:}(T\ t), u{:}(Bx\ t\ v) \vdash_{\Sigma_\Box(\mathbf{NS4})} b_1{:}(Bx\ t'\ p_1)$, and hence

$$\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} t'' : \prod_{v:(T\ t)} \prod_{u:(Bx\ t\ v)} (Bx\ t'\ p_1),$$

where $t'' \stackrel{\text{def}}{=} \lambda v{:}(T\ t)\lambda u{:}(Bx\ t\ v).b_1$. Then, finally

$$\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} (Bx_{\supset\text{-}\mathbf{I}}\ t\ t'\ (\lambda v{:}(T\ t).p_1)t'') : (Bx\ (\supset\ t\ t')\ (\supset\text{-I}\ t\ t'\ (\lambda v{:}(T\ t).p_1))).$$

•$p \equiv (\supset\text{-E}\ t\ t' p_1\ p_2) : (T\ t')$: since p is well-typed, $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} p_1 : (T\ (\supset\ \ t\ t'))$ and $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} p_2 : (T\ t)$. Since $FV(p_1), FV(p_2) \subseteq FV(p)$, then both $C(p_1, \gamma_\Box(\Delta))$ and $C(p_2, \gamma_\Box(\Delta))$ hold. Then, by IH there exist b_1, b_2 such that $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} b_1 : (Bx\ (\supset\ \ t\ t')\ p_1)$ and $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} b_2 : (Bx\ t\ p_2)$. Therefore $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} (Bx_{\supset\text{-E}}\ t\ t'\ p_1\ p_2\ b_1\ b_2) : (Bx\ t'\ (\supset\text{-E}\ t\ t'\ p_1\ p_2))$.

•$p \equiv (\supset_\Box\text{-I}\ t\ t'\ (\lambda v{:}(T\ \Box t)\lambda u{:}(Bx\ \Box t\ v).p_1) : (T\ (\supset\ \ \Box t\ t')))$: since p is well-typed, $\Gamma_X, \gamma_\Box(\Delta), v{:}(T\ \Box t), u{:}(Bx\ \Box t\ v) \vdash_{\Sigma_\Box(\mathbf{NS4})} p_1 : (T\ t')$. Moreover, the property

$$C\left(p_1, (\gamma_\Box(\Delta), v{:}(T\ \Box t), u{:}(Bx\ \Box t\ v))\right)$$

holds. Then by IH there exists b_1 such that $\Gamma_X, \gamma_\Box(\Delta), v{:}(T\ \Box\ t), u{:}(Bx\ \Box t\ v) \vdash_{\Sigma_\Box(\mathbf{NS4})} b_1 : (Bx\ t'\ p_1)$. By abstracting we obtain

$$\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} t'' : \prod_{v:(T\ \ \Box t)} \prod_{u:(Bx\ \ \Box t\ \ v)} (Bx\ t'\ p_1).$$

where $t'' \stackrel{\text{def}}{=} \lambda v{:}(T\ \Box t)\lambda u{:}(Bx\ \Box t\ v).b_1$. Then

$$\begin{aligned}\Gamma_X, \gamma_\Box(\Delta) \quad & \vdash_{\Sigma_\Box(\mathbf{NS4})} (Bx_{\supset_\Box\text{-I}}\ t\ t'(\lambda v{:}(T\ \Box t).\lambda u{:}(Bx\ \Box t\ v).p_1)(t'')) : \\ & (Bx\ (\supset\ \ \Box t\ t')\ (\supset_\Box\text{-I}\ t\ t'\ (\lambda v{:}(T\ \Box t)\lambda u{:}(Bx\ \Box t\ \ v).p_1))).\end{aligned}$$

□

A.3.7 Proof of Lemma 5.7

By induction on the structure of π.
Base Step: φ is an assumption, i.e. $\varphi \in \Delta$. Then we take $p = v_\varphi \in dom(\gamma_\Box(\Delta))$.
Inductive Step: by cases on the last rule applied. We see only some significant cases, the other being similar.

•$\pi \equiv \supset\text{-I}_{\psi\theta}(\pi')$: then $(X, (\Delta, \psi)) \models_{\mathbf{NS4}} \pi' : \theta$ and hence by IH there exists a canonical term p such that $\Gamma_X, \gamma_\Box(\Delta, \psi) \vdash_{\Sigma_\Box(\mathbf{NS4})} p : (T\ \varepsilon_X(\theta))$. Now there are two cases, depending on whether ψ is boxed or not.

- if ψ is boxed, then $\gamma_\Box(\{\psi\}) = v{:}(T\ \varepsilon_X(\psi)), u{:}(Bx\ \varepsilon_X(\psi)\ v)$. Then

  $$\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} t : \prod_{v:(T\ \ \varepsilon_X(\psi))} \prod_{u(Bx\ \ \varepsilon_X(\psi)\ \ v)} (T\ \varepsilon_X(\theta))$$

  where $t \stackrel{\text{def}}{=} \lambda v{:}(T\ \varepsilon_X(\psi)).\lambda u{:}(Bx\ \varepsilon_X(\psi)\ v).p$. Hence
  $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} (\supset_\Box\text{-I}\ \varepsilon_X(\psi)\ \varepsilon_X(\theta)\ t){:}(T\ (\supset \varepsilon_X(\varphi)\ \varepsilon_X(\theta)))$
- otherwise, ψ is not boxed; then $\gamma_\Box(\{\psi\}) = v{:}(T\ \varepsilon_X(\psi))$. Then $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} (\lambda v{:}(T\ \varepsilon_X(\psi)).p) : (\Pi v{:}(T\ \varepsilon_X(\psi))).(T\ \varepsilon_X(\theta))$. Hence immediately
  $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} (\supset\text{-I}\ \varepsilon_X(\psi)\varepsilon_X(\theta)(\lambda v{:}(T\varepsilon_X(\psi)).p)) : (T(\supset\ \ \varepsilon_X(\psi)\ \varepsilon_X(\theta)))$.

•$\pi \equiv \Box\text{-I}_\psi(\pi')$; then $(X, \Box\Delta) \models_{\mathbf{NS4}} \pi' : \psi$. By IH there exists a canonical term p_1 such that $\Gamma_X, \gamma_\Box(\Box\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} p_1 : (T\ \varepsilon_X(\psi))$. Since $C(p_1, \gamma_\Box(\Box\Delta))$ always holds, by Lemma 5.6 there exists a canonical term b_1 such that $\Gamma_X, \gamma_\Box(\Box\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} b_1 : (Bx\ \varepsilon_X(\psi)\ p_1)$. Hence $\Gamma_X, \gamma_\Box(\Box\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} (\Box\text{-I}\ \varepsilon_X(\psi)\ p_1\ b_1) : (T\ \Box\varepsilon_X(\psi))$.

•$\pi \equiv \supset\text{-E}_{\psi,\varphi}(\pi', \pi'')$: then $(X, \Delta) \models_{\mathbf{NS4}} \pi' : \psi$ and $(X, \Delta) \models_{\mathbf{NS4}} \pi'' : \psi \supset \varphi$. Therefore by IH there exist two canonical terms p_1, p_2 such that $\Gamma_X, \gamma_\Box(\Box\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} p_1 : (T\ \varepsilon_X(\psi))$ and $\Gamma_X, \gamma_\Box(\Box\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} p_2 : (T\ \varepsilon_X(\psi \supset \varphi))$. Then, $\Gamma_X, \gamma_\Box(\Box\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} (\supset\text{-E}\ \varepsilon_X(\psi)\ \varepsilon_X(\varphi)\ p_2\ p_1) : (T\varepsilon_X(\varphi))$ □

A.3.8 Proof of Theorem 5.8

The result follows immediately from Lemma 5.7 and the following two technical Lemma A.2, A.3. For sake of simplicity, we adopt the following definitions: for p term and Γ context

$$C'(p,\Gamma) \stackrel{\text{def}}{=} \text{ for all } c \in \mathrm{FV}(p), \text{ for all } (c{:}(T\ t)) \in \Gamma, \text{ there exists } (b{:}(Bx\ t\ c)) \in \Gamma$$
$$\alpha_p(\Gamma) \stackrel{\text{def}}{=} \{\varphi \mid (v{:}(T\ \varepsilon_X(\varphi))) \in \Gamma \text{ and } v \in \mathrm{FV}(p)\}.$$

Intuitively, the set $\alpha_p(\Gamma)$ contains the "active assumptions" in the context Γ for p.

Lemma A.2 *If there is a canonical term b such that $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} b : (Bx\ \varphi\ p)$ then $C'(p, \gamma_\Delta(\Delta))$ holds.*

Proof. By lemma of characterization, a canonical form d of type $(Bx\ \varphi\ p)$ must be $\zeta M_1 \ldots M_k$, where k is the arity of ζ. By inspection of $\Sigma_\Box(\mathbf{NS4})$ and $\Gamma_X, \gamma_\Box(\Delta)$ we see that the only choices are $\zeta \in \{u_\varphi, Bx_{\supset\text{-I}}, Bx_{\supset\text{-E}}, Bx_{\Box\text{-I}}, Bx_{\Box\text{-E}}, \ldots\}$.
Base Step: $b = u_\varphi \in \gamma_\Box(\Delta)$; then, immediately, $v_\varphi \in \gamma_\Box(\Delta)$ and hence $C'(p, \gamma_\Box(\Delta))$ holds.
Inductive Step: we see some significant cases, the other being similar.

•$b \equiv (Bx_{\supset\text{-I}}\ t\ t'(\lambda v{:}(T\ t).p_1)(\lambda v{:}(T\ t)\lambda u{:}(Bx\ t\ v).b_1))$: then $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} b : (Bx\ (\supset\ t\ t')\ p)$. Since b is well-typed we have that

$$\Gamma_X, \gamma_\Box(\Delta), v : (T\ t), u : (Bx\ t\ v) \vdash_{\Sigma_\Box(\mathbf{NS4})} b_1 : (Bx\ t'\ p_1).$$

By IH $C'(p_1, (\gamma_\Box(\Delta), v : (T\ t), u : (Bx\ t\ v)))$. Since $p \equiv \supset\text{-I}\ t\ t'(\lambda v : (T\ t).p_1)$, we have $\mathrm{FV}(p) = \mathrm{FV}(p_1) \setminus \{v\}$, therefore $C'(p, \gamma(\Delta))$ holds.

•$b \equiv (Bx_{\supset\text{-E}}\ t\ t'\ p_1\ p_2\ b_1\ b_2)$: then $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} b : (Bx\ t'\ p)$ where $p \stackrel{\text{def}}{=} (\supset\text{-E}\ t\ t'\ p_1\ p_2)$. Since b is well-typed we have that $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} b_1 : (Bx\ (\supset\ t\ t')\ p_1)$ and $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} b_2 : (Bx\ t\ p_2)$. By IH, $C'(p_1, \gamma_\Box(\Delta))$ and $C'(p_2, \gamma_\Box(\Delta))$ hold; then $C'(p, \gamma_\Box(\Delta))$ holds, since $\mathrm{FV}(p_1) \cup \mathrm{FV}(p_2) = \mathrm{FV}(p)$.

•$b \equiv (Bx_{\supset_\Box\text{-I}}\ t\ t'\ p_1\ p_2)$, where $p_1 \stackrel{\text{def}}{=} \lambda v{:}(T\ \Box t).\lambda u{:}(Bx\ \Box t\ v).p_1'$ and $p_2 \stackrel{\text{def}}{=} \lambda v : (T\ \Box t).\lambda u{:}(Bx\ \Box t\ v).b_1$. Then $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} b : (Bx\ (\supset\ \Box t\ t')\ p)$ where $p \equiv (\supset_\Box\text{-I}\ t\ t'\ p_1)$. Since b is well-typed we have that

$$\Gamma_X, \gamma_\Box(\Delta), v : (T\ \Box t), u : (Bx\ \Box t\ v) \vdash_{\Sigma_\Box(\mathbf{NS4})} b_1{:}(Bx\ t'\ p_1').$$

By IH, $C'(p_1', (\gamma_\Box(\Delta), v{:}(T\ \Box t), u : (Bx\ \Box t\ v)))$ holds, and then $C'(p_1, \gamma_\Box(\Delta))$ holds too. Therefore $C'(p, \gamma_\Box(\Delta))$ holds.

•$b \equiv (Bx_{\Box\text{-I}}\ t\ p_1\ b_1)$: then $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} b : (Bx\ (\Box t)\ p)$ with $p \equiv (\Box\text{-I}\ t\ p_1\ b_1)$. Since b is well-typed we have $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} b_1 : (Bx\ t\ p_1)$, and hence by IH $C'(p_1, \gamma(\Delta))$ holds. Therefore $C'(p, \gamma(\Delta))$ holds as well, because the free variables in b_1 are typed in $\gamma(\Delta)$ only by the Bx judgement. □

Lemma A.3 *Given a canonical term p such that $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} p : (T\ t)$, there exists a proof π such that $(X, \alpha_p(\gamma_\Box(\Delta))) \models_{\mathbf{NS4}} \pi : \delta_X(t)$.*

Proof. By lemma of characterization, a canonical form p of type $(T\ t)$ must be $\zeta M_1 \ldots M_k$, where k is the arity of ζ. By inspection of $\Sigma_\Box(\mathbf{NS4})$ e $\Gamma_X, \gamma_\Box(\Delta)$ we see that the only choices for ζ are $\zeta \in \{v_\varphi, \supset\text{-I}, \supset\text{-E}, \Box\text{-I}, \Box\text{-E}, \supset_\Box\text{-I} \ldots\}$. We proceed by induction.
Base Step: $p = v$. Then, $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} p : (T\ t)$ and moreover we have that $\alpha_p(\gamma_\Box(\Delta)) = \{\delta_X(t)\}$. Taken $\pi = \delta_X(t)$, we obtain $(X, \delta_X(t)) \models_{\mathbf{NS4}} \pi : \delta_X(t)$.
Inductive Step. We see only some significant cases.

•$p \equiv (\supset\text{-E}\ t\ t'\ p_1\ p_2)$: then $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} p : (T\ t')$. Since p is well-typed we have that $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} p_1{:}(T(\supset\ t\ t'))$ and $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} p_2 : (T\ t)$. By IH there exist π', π'' such that $(X, \alpha_{p_1}(\gamma_\Box(\Delta))) \models_{\mathbf{NS4}} \pi' : \delta_X(\supset t\ t')$ and $(X, \alpha_{p_2}(\gamma_\Box(\Delta))) \models_{\mathbf{NS4}} \pi'' : \delta_X(t)$. Then taken $\pi = \supset\text{-E}_{\delta_X(t),\delta_X(t')}(\pi', \pi'')$ we obtain that $(X, \alpha_p(\gamma_\Box(\Delta))) \models_{\mathbf{NS4}} \pi : \delta_X(t')$.

•$p \equiv (\supset\text{-I}\ t\ t'(\lambda v : (T\ t).p_1))$: then, $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} p : (T(\supset t\ t'))$. Since p is well-typed we have that $\Gamma_X, \gamma_\Box(\Delta), v{:}(T\ t) \vdash_{\Sigma_\Box(\mathbf{NS4})} p_1 : (T\ t')$
(i.e. $\Gamma_X, \gamma_\Box(\Delta, \delta_X(t)) \vdash_{\Sigma_\Box(\mathbf{NS4})} p_1 : (T\ t')$ because $\gamma_\Box(\Delta, \delta_X(t)) \supseteq \gamma_\Box(\Delta), v : (T\ t)$). By IH there exists π' such that $(X, \alpha_{p_1}(\gamma_\Box(\Delta, \delta_X(t)))) \models_{\mathbf{NS4}} \pi'{:}\delta_X(t')$. Moreover,
$\alpha_{p_1}(\gamma_\Box(\Delta, \delta_X(t))) \subseteq \alpha_p(\gamma(\Delta)), \delta_X(t)$ since $\mathrm{FV}(p_1) \subseteq \mathrm{FV}(p) \cup \{v\}$. Then
$(X, \alpha_p(\gamma_\Box(\Delta)), \delta_X(t)) \models_{\mathbf{NS4}} \pi'{:}\delta_X(t')$; taken $\pi = \supset\text{-I}_{\delta_X(t),\delta_X(t')}(\pi')$, we finally obtain $(X, \alpha_p(\gamma_\Box(\Delta))) \models_{\mathbf{NS4}} \pi : \delta_X(\supset t\ t')$.

•$p \equiv (\supset_\Box\text{-I}\ t\ t'\ (\lambda v{:}(T\ t)\lambda u{:}(Bx\ t\ v).p_1))$: then $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} p : (T\ (\supset t\ t'))$. Since p is well-typed we have

$$\Gamma_X, \gamma_\Box(\Delta), v{:}(T\ t), u{:}(Bx\ t\ v) \vdash_{\Sigma_\Box(\mathbf{NS4})} p_1 : (T\ t').$$

Moreover, since $\delta_X(t)$ is boxed, it is $\Gamma_X, \gamma_\Box(\Delta, \delta_X(t)) \vdash_{\Sigma_\Box(\mathbf{NS4})} p_1 : (T\ t')$. By IH there exists π' such that $(X, \alpha_{p_1}(\gamma_\Box(\Delta, \delta_X(t)))) \models_{\mathbf{NS4}} \pi'{:}\delta_X(t')$. Now, $\alpha_{p_1}(\gamma_\Box(\Delta, \delta_X(t))) \subseteq \alpha_p(\gamma_\Box(\Delta)) \cup \{\delta_X(A)\}$, since $\mathrm{FV}(p_1) \subseteq \mathrm{FV}(p) \cup \{v, u\}$. Then $(X, \alpha_p(\gamma_\Box(\Delta)), \delta_X(t)) \models_{\mathbf{NS4}} \pi' : \delta_X(t')$. Taken $\pi = \supset_\Box\text{-I}_{\delta_X(t),\delta_X(t')}(\pi')$ we obtain that $(X, \alpha_p(\gamma_\Box(\Delta))) \models_{\mathbf{NS4}} \pi : \delta_X(\supset t\ t')$.

•$p \equiv (\Box\text{-I}\ t\ p_1\ b_1)$: then $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} p : (T\ (\Box t))$. Since p is well-typed we have $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} b_1 : (Bx\ t\ p_1)$ and $\Gamma_X, \gamma_\Box(\Delta) \vdash_{\Sigma_\Box(\mathbf{NS4})} p_1 : (T\ t)$. By IH there exists π' such that $(X, \alpha_{p_1}(\gamma(\Delta))) \models_{\mathbf{NS4}} \pi'{:}\delta_X(t)$. By the lemma A.2, $C'(p_1, \gamma_\Box(\Delta))$ holds. Now, for each $\psi \in \alpha_{p_1}(\gamma_\Box(\Delta))$, by definition of α there is an assumption $(v{:}(T\ \varepsilon_X(\psi))) \in \gamma_\Box(\Delta)$ such that $v \in \mathrm{FV}(p_1)$. Since $C'(p_1, \gamma_\Box(\Delta))$ holds, we have that there is an assumption $(u : (Bx\ \varepsilon_X(\psi)\ v)) \in \gamma_\Box(\Delta)$, but by definition of $\gamma_\Box$, this means that ψ is boxed. Then $\alpha_{p_1}(\gamma_\Box(\Delta))$ contains only boxed formulæ. Then we can take $\pi = \Box\text{-I}(\pi')$ obtaining $(X, \alpha_p(\gamma_\Box(\Delta))) \models_{\mathbf{NS4}} \pi : \Box\delta_X(t)$. □

From Lemma A.3 follows the definition of the decoding function $\delta_{X,\Delta}^{\Sigma_\Box(\mathbf{NS4})}$:

$$\delta_{X,\Delta}^{\Sigma_\Box(\mathbf{NS4})}(\varphi) \stackrel{\text{def}}{=} \delta_{X,\Delta;\emptyset}^{\Sigma_\Box(\mathbf{NS4})}(\varphi)$$
$$\delta_{X,\Delta;\Psi}^{\Sigma_\Box(\mathbf{NS4})}(v) \stackrel{\text{def}}{=} \delta_X(t) \qquad \textit{for}\ (v{:}(T\ t)) \in \gamma_\Box(\Delta) \cup \Psi.$$
$$\delta_{X,\Delta;\Psi}^{\Sigma_\Box(\mathbf{NS4})}(\supset\text{-E}\ t\ t'\ p_1\ p_2) \stackrel{\text{def}}{=} \supset\text{-E}_{\delta_X(t),\delta_X(t')}(\delta_{X,\Delta;\Psi}^{\Sigma_\Box(\mathbf{NS4})}(p_1), \delta_{X,\Delta;\Psi}^{\Sigma_\Box(\mathbf{NS4})}(p_2))$$
$$\delta_{X,\Delta;\Psi}(\supset\text{-I}\ t\ t'(\lambda v{:}(T\ t).p)) \stackrel{\text{def}}{=} \supset\text{-I}_{\delta_X(t),\delta_X(t')}(\delta_{X,\Delta;\Psi,(v:(T\ t))}^{\Sigma_\Box(\mathbf{NS4})}(p))$$
$$\delta_{X,\Delta;\Psi}^{\Sigma_\Box(\mathbf{NS4})}(\supset_\Box\text{-I}\ t\ t'\ (\lambda v{:}(T\ t)\lambda u{:}(Bx\ t\ v).p)) \stackrel{\text{def}}{=} \supset_\Box\text{-I}_{\delta_X(t),\delta_X(t')}(\delta_{X,\Delta;\Psi,(v:t)}^{\Sigma_\Box(\mathbf{NS4})}(p))$$
$$\delta_{X,\Delta;\Psi}^{\Sigma_\Box(\mathbf{NS4})}(\Box\text{-I}\ t\ p\ b) \stackrel{\text{def}}{=} \Box\text{-I}(\delta_{X,\Delta;\Psi}^{\Sigma_\Box(\mathbf{NS4})}(p))$$
$$\delta_{X,\Delta}(\Box\text{-E}\ t\ p) \stackrel{\text{def}}{=} \Box\text{-E}(\delta_{X,\Delta;\Psi}^{\Sigma_\Box(\mathbf{NS4})}(p))$$

A.3.9 Proof of Theorem 6.1

Hints:

- $1 \Rightarrow 2$: by induction on t.
- $2 \Rightarrow 3$: by induction on n.
- $3 \Rightarrow 1$: by induction on v. Alternatively, replace $(V\ \varepsilon_X(\psi))$ by $\prod_{w:U}(T\ w\ \varepsilon_X(\psi))$, and $(T\ \varepsilon_X(\psi))$ by $(T\ w\ \varepsilon_X(\psi))$, everywhere.

- $3 \Rightarrow 2$: it is possible to express V in terms of Na, by means of Σ-types: $(V\ \varepsilon_X(\varphi)) = \sum_{x:(T\ \varepsilon_X(\varphi))}(Cl\ \varepsilon_X(\varphi)\ x)$. Hence, proof of $(V\ \varepsilon_X(\varphi))$ is, a proof of $(T\ \varepsilon_X(\varphi))$ together with the proof that it does not depend on any assumptions. This is not possible in LF but it is in some higher-order logical framework, such as CIC. □

A.3.10 Proof of Theorem 6.2

Similar to Theorem 6.1.

References

[1] S. Abramsky, D. Gabbay, and T. Maibaum, editors. *Handbook of Logic in Computer Science*. Oxford University Press, 1992.

[2] A. Avron. Simple consequence relations. *Infor. Comp.*, 92:105–139, Jan. 1991.

[3] A. Avron, F. Honsell, I. A. Mason, and R. Pollack. Using Typed Lambda Calculus to implement formal systems on a machine. *J. Automated Reasoning*, 9:309–354, 1992.

[4] H. Barringer, J. H. Cheng, and C. B. Jones. A logic covering undefiness in program proofs. *Acta Informatica*, 21:251–269, 1984.

[5] D. Basin, S. Matthews, and L. Viganó. A modular presentation of modal logics in a logical framework. This volume.

[6] M. Coen. *Interactive Program Derivation*. PhD thesis, University of Cambridge, 1992.

[7] T. Coquand and G. Huet. The calculus of constructions. *Information and Control*, 76:95–120, 1988.

[8] C. Cornes, J. Courant, J.-C. Fillâtre, G. Huet, P. Manoury, C. Muñoz, C. Murthy, C. Parent, C. Paulin-Mohring, A. Saïbi, and B. Werner. *The Coq Proof Assistant Reference Manual - Version 5.10*. INRIA, Rocquencourt, July 1995. Available at `ftp://ftp.inria.fr/INRIA/coq/V5.10/doc/Reference-Manual.dvi.Z`.

[9] M. Fitting. *Proof Methods for Modal and Intuitionistic Logic*, volume 109 of *Synthese Library*. Reidel, 1983.

[10] M. Fitting. Tableaus for many-valued modal logic. *Studia Logica*, 55:63–87, 1995.

[11] M. Fitting. leanT^AP revisited. Unpublished notes, Mar. 1997. Available at `ftp://gillet3.lehman.cuny.edu/pub/fitting/leantap.ps.gz`

[12] P. Gardner. Equivalence between logics and their representing type theories. *Mathematical Structures in Computer Science*, 5:323–349, 1995.

[13] G. Gentzen. Investigations into logical deduction. In M. Szabo, editor, *The collected papers of Gerhard Gentzen*, pages 68–131. North Holland, 1969.

[14] D. Harel. Dynamic logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume II, pages 497–604. Reidel, 1984.

[15] R. Harper, F. Honsell, and G. Plotkin. A framework for defining logics. *J. ACM*, 40(1):143–184, Jan. 1993.

[16] M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of ACM*, 32:137–162, 1985.

[17] F. Honsell and M. Miculan. A natural deduction approach to dynamic logics. In S. Berardi and M. Coppo, editors, *Proc. of TYPES'95*, LNCS number 1158, pages 165–182, Turin, Mar. 1995. Springer-Verlag, 1996. A preliminar version has been communicated to the *TYPES'94 Annual Workshop*, Båstad, July 1994.

[18] G. E. Hughes and M. J. Cresswell. *A companion to Modal Logic*. Methuen, 1984.

[19] Z. Luo, R. Pollack, and P. Taylor. *How to use LEGO (A Preliminary User's Manual)*. Department of Computer Science, University of Edinburgh, Oct. 1989.

[20] P. Martin-Löf. On the meaning of the logical constants and the justifications of the logic laws. Technical Report 2, Scuola di Specializzazione in Logica Matematica, Dipartimento di Matematica, Università di Siena, 1985.

[21] S. Martini and A. Masini. A computational interpretation of modal proofs. In H. Wansing, editor, *Proof theory of Modal Logics*. Kluwer, 1994.

[22] S. Merz. Mechanizing TLA in Isabelle. In R. Rodošek, editor, *Workshop on Verification in New Orientations*, pages 54–74, Maribor, July 1995. Univ. of Maribor.

[23] M. Miculan. The expressive power of structural operational semantics with explicit assumptions. In H. Barendregt and T. Nipkow, editors, *Proceedings of TYPES'93*, LNCS number 806, pages 292–320. Springer-Verlag, 1994.

[24] M. Miculan. *Encoding Logical Theories of Programs*. PhD thesis, Dipartimento di Informatica, Università di Pisa, Italy, Mar. 1997.

[25] B. Nordström, K. Petersson, and J. M. Smith. Martin-Löf's type theory. In Abramsky et al. [1].

[26] F. Pfenning and H.-C. Wong. On a modal λ-calculus for S4. In *Proc. MFPS'95*, 1995.

[27] D. Prawitz. *Natural Deduction*. Almqvist & Wiksell, Stockholm, 1965.

[28] P. Schroeder-Heister. A natural extension of natural deduction. *J. Symbolic Logic*, 49(4):1284–1300, 1984.

[29] A. Simpson. *The Proof Theory and Semantics of Intuitionistic Modal Logic*. PhD thesis, University of Edinburgh, 1993.

[30] C. Stirling. Modal and Temporal Logics. In Abramsky et al. [1], pages 477–563.

[31] J. van Benthem. *Modal logic and classical logic*, volume 3 of *Monographs in philosophical logic and formal linguistics*. Bibliopolis, Napoli, 1983.

[32] L. A. Wallen. *Automated Proof Search in Non-Classical Logics*. MIT Press, 1990.

[33] B. Werner. *Une théorie des constructions inductives*. PhD thesis, Univ. Paris 7, 1994.