Introspective Kripke models and normalisation by evaluation for the $\lambda^{\Box}$-calculus

Miëtek Bak
mietek@bak.io

We present a class of Kripke-style models that is sound and complete for the necessity-only fragment of the intensional modal logic S4. Our soundness and completeness proofs are constructive, and their composition yields a normalisation by evaluation procedure for the Pfenning-Davies system of natural deduction, known as the $\lambda^{\Box}$-calculus.

The entire development is formalised in AGDA, a dependently typed functional programming language, and available online at http://github.com/mietek/imla2017.

Introduction

We seek a *total functional programming* language, as defined by Turner[32], which would approach the expressivity of McCarthy's[23] LISP in the domain of meta-programming. We view the present work as a stepping stone towards meeting Sheard's[30] challenge of *intensional analysis of syntax* in a total setting. To this end, we revisit the $\lambda^{\Box}$-calculus of Pfenning and Davies[28], choosing to interpret $\Box$ as a *quotation modality*.

It is important to note that the Pfenning-Davies calculus is subtly different from the system given by Bierman and de Paiva[7], even though both correspond to the intuitionistic modal logic S4. The convertibility relation supplied with the $\lambda^{\Box}$-calculus is not congruent under the $\Box$-introduction rule:

$$\frac{\varnothing\,;\Delta \vdash A}{\Gamma\,;\Delta \vdash \Box A}$$

In other words, wrapping a λ-term with the above rule inhibits evaluation, which is also the precise purpose of the LISP special form, `quote`. This connection validates our interpretation of $\Box$ as a quotation modality. Therefore, we denote the $\Box$ introduction rule **quot**.[1]

Normalisation by evaluation (NbE) is a technique for computing the normal form of λ-terms, introduced by Martin-Löf[22], rediscovered by Berger and Schwichtenberg[6], and identified as the computational content of the composition of soundness and completeness proofs by C. Coquand[10]. Assuming a suitable semantics, NbE can be understood as the evaluation of a given λ-term, or the *reflection* of the term into the model, followed by the *reification* of the resulting value back to a term in normal form. Abel[1] provides an in-depth discussion.

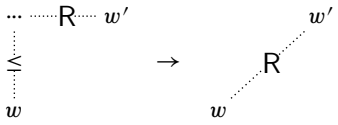
Under the principle of *propositions as types,* recently surveyed by Wadler[33], a constructive proof of soundness corresponds to an *evaluation function* on derivations, `reflect` M. Similarly, a constructive completeness proof corresponds to an *inverse of the evaluation function* on semantic objects, `reify` o.

To the best of our knowledge, no published class of models for an S4-style modal logic supports a constructive proof of completeness. The proofs given by Ono[27], Božić and Došen[8], Fischer Servi[15], Ewald[14], Plotkin and Stirling[29], Wijesekera[34], Simpson[31], and Alechina *et al.*[2] all proceed by

[1] Unfortunately, `quote` is a reserved identifier in many languages, including AGDA.

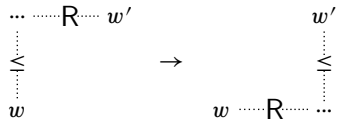
Submitted to:
IMLA 2017

Persistence.



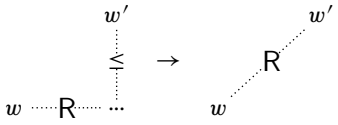
$$w \leq_{\S} \mathsf{R}\ w' \rightarrow w\ \mathsf{R}\ w'$$

Minor persistence.



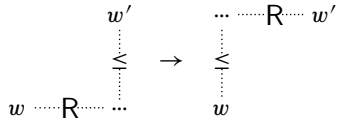
$$w \leq_{\S} \mathsf{R}\ w' \rightarrow w\ \mathsf{R}_{\S}\leq w'$$

Brilliance.



$$w\ \mathsf{R}_{\S}\leq w' \rightarrow w\ \mathsf{R}\ w'$$

Minor brilliance.



$$w\ \mathsf{R}_{\S}\leq w' \rightarrow w \leq_{\S} \mathsf{R}\ w'$$

Figure 1. Conditions in Kripke-style models.

contradiction. Without a constructive completeness proof, or a reify o function, a NbE procedure cannot be defined.

One of the problems with supplying a complete semantics for the $\lambda^{\Box}$-calculus follows from the difficult character of the $\Box$-elimination rule, which we denote **eval**:

$$\frac{\Gamma\,;\Delta \vdash \Box A \quad \Gamma\,;\Delta, A \vdash C}{\Gamma\,;\Delta \vdash C}$$

The above rule belongs to a class of *general elimination* (GE) rules, after Dyckhoff[13], together with the usual rules for the elimination of disjunction, absurdity, and existential quantification. GE rules are distinguished by the presence of a proposition C with no structural link to the proposition being eliminated. GE rules are also described by Girard *et al.*[17] as *'catastrophic'*, *'atrocities'*, and *'very bad'*.

Nevertheless, Ilik[20] provides a complete semantics for all connectives of propositional logic, whether well-behaved or not. Ilik's solution, which we apply in our work, is a *continuation-passing style* transformation of the semantics, based on Danvy's[12] *type-directed partial evaluation.*

Our models build on the standard *possible worlds semantics* for intuitionistic logic, introduced by Kripke[21], which includes an *intuitionistic accessibility relation* between worlds, $w \leq w'$. As usual for modal logics, we also include a *modal accessibility relation,* $w\ \mathsf{R}\ w'$.

Most models for S4-like modal logics, as surveyed by Iemhoff[19], specify at least one condition that explains the interaction between the two relations. Some of these conditions are summarised in Figure 1. We require that intuitionistic accessibility implies modal accessibility, and denote this condition *vindication,* since it follows directly from both *brilliance* and *persistence*.

The main contribution of the present work is a novel *introspection condition,* which allows us to embed closed, non-normal form derivations in the semantics, and thereby explain the meaning of the $\Box$ modality. This condition also suffices to avoid having to extend our models with *exploding worlds*, in contrast with Ilik's work.

Related work

Our embedding of syntax into the model is superficially similar to the *glueing construction* of T. Coquand and Dybjer[11], who offer a semantics for combinatory logic. There is a stronger connection with Gabbay's and Nanevski's[16] semantics for contextual modal type theory, given by Nanevski *et al.*[25], although our interpretation of the $\Box$ modality is different from theirs.

The strongest connection of all is found through Artemov's[5] work on the *logic of proofs* (LP), which provides an answer to Gödel's[18] question about the intended semantics for S4. Originally, the $\Box$ modality was meant to be interpreted as a provability predicate, with $\Box A$ meaning *"there exists a proof of A"*. Artemov argues that the failure of Gödel's provability interpretation is due to the non-constructive character of the existential quantifier. We support this point of view, and observe that it is precisely the availability of a strong notion of existence in our constructive meta-language which allows us to justify a new reading of the $\Box$ modality.

If the standard intuitionistic Kripke semantics corresponds to the McKinsey-Tarski[24] embedding of intuitionistic logic in S4, our semantics corresponds to the Artemov embedding of S4 in LP.

Notation

In this paper, we adopt an informal dialect of the AGDA dependently typed functional programming language, originally due to Norell[26]. Inessential technical details, such as those relating to universe polymorphism, are omitted. We make no such concessions in our full formal development.

All meta-variables are implicitly universally quantified. Any omitted proof can be recovered by means of a routine inductive argument, unless otherwise noted.

The function space, or meta-level implication, is written as $X \to Y$, introduced through λ-expressions, $f = \lambda\, x\, .\, y$, and eliminated through juxtaposition, $f\ x$. We refer to the identity function by id. Function composition is written as $f \circ g$.

By convention, the letters A, B, C refer to propositions, and Γ, Δ, Ξ refer to collections of propositions. Proof trees, or derivations, are referred to by M, N, and continuations by κ.

Outline

Our argument is structured as follows:

(1) We begin by formalising a calculus of proof trees as an indexed type family, or a *syntactic entailment relation,* $\Gamma\,;\Delta \vdash A$, where each proof tree represents a term-in-context. Normal forms are defined in a similar manner.

(2) We give a class of *introspective Kripke models*, which include intuitionistic accessibility, $w \leq w'$, modal accessibility, $w\ \mathsf{R}\ w'$, vindication, $w \leq w' \to w\ \mathsf{R}\ w'$, and introspection, $w \to \Gamma\,;\Delta$. We define a *forcing relation*, $w \Vdash A$, in order to explain the meaning of a given proposition being true in a particular world.

(3) We define a *semantic entailment relation,* $\Gamma\,;\Delta \vDash A$, as forcing in all worlds of all models. Soundness of the semantics with respect to the calculus is shown by defining an *evaluation function,* $\mathsf{reflect}\ M$, which maps proof trees to semantic objects, or values within the model.

(4) We give a *canonical model* that identifies worlds with contexts. Defining an *inverse of the evaluation function,* $\mathsf{reify}\ o$, shows completeness of the semantics with respect to the calculus. Finally, we obtain a *normalisation function,* $\mathsf{nbe}\ M$.

1 Syntax

1.1 Types

Let us consider a set of types, Type, that corresponds to the connectives of propositional modal logic S4, and includes a base type $\bullet$, a function type $A \Rightarrow B$, and a type of quotations $\Box A$:

$$\frac{}{\bullet : \mathsf{Type}} \qquad \frac{A : \mathsf{Type} \quad B : \mathsf{Type}}{A \Rightarrow B : \mathsf{Type}} \qquad \frac{A : \mathsf{Type}}{\Box A : \mathsf{Type}}$$

The constructions described in the present work can be easily generalised to product and sum types, Girard's disdain notwithstanding. This extension, omitted here for lack of space, is included in our full formal development.

1.2 Contexts

We define contexts, $\mathsf{Context} = \mathsf{Stack}^2\,\mathsf{Type}\,\mathsf{Type}$, as type stack pairs, or pairs of type stacks. To rephrase: the first element of a given context is a stack of types representing the non-modal assumptions, and the second element is a stack of types representing the modal assumptions.

A stack is another name for a list, written in a more natural, reverse order. An empty stack is constructed as $\varnothing$. We overload the same symbol to mean the constructor of the unit type, $\top$. Pushing an element on top of a stack is written as $\Gamma\,, A$.

$$\frac{}{\varnothing : \mathsf{Stack}\,X} \qquad \frac{\Gamma : \mathsf{Stack}\,X \quad A : X}{\Gamma\,, A : \mathsf{Stack}\,X}$$

Stack pairs are constructed as $\Gamma\,;\Delta$, as opposed to regular pairs, which are constructed as $x\,, y$. We allow the first and second element of any pair to be accessed with the first and second projection function, $\pi_1\ Z$ and $\pi_2\ Z$, respectively.

Instead of explicit variable names, our formalisation of the $\lambda^{\Box}$-calculus uses de Bruijn indices, formalised as an inductive stack membership relation, $A \in \Gamma$. The letters i, j refer to indices.

$$\frac{}{\mathbf{top} : A \in \Gamma\,, A} \qquad \frac{i : A \in \Gamma}{\mathbf{pop}\ i : A \in \Gamma\,, B}$$

Following Chapman[9], we define an inclusion relation on stacks, $\Gamma \subseteq \Gamma'$, as an inductive type of *order-preserving embeddings*. The letters η, μ refer to stack inclusions.

$$\frac{}{\mathbf{done} : \varnothing \subseteq \varnothing} \qquad \frac{\eta : \Gamma \subseteq \Gamma'}{\mathbf{skip}\ \eta : \Gamma \subseteq \Gamma'\,, A} \qquad \frac{\eta : \Gamma \subseteq \Gamma'}{\mathbf{keep}\ \eta : \Gamma\,, A \subseteq \Gamma'\,, A}$$

Inclusion on stacks is reflexive, transitive, and has a minimal element:

$$\begin{aligned} \mathsf{refl}_{\subseteq} &: \Gamma \subseteq \Gamma \\ \mathsf{trans}_{\subseteq} &: \Gamma \subseteq \Gamma' \to \Gamma' \subseteq \Gamma'' \to \Gamma \subseteq \Gamma'' \\ \mathsf{bot}_{\subseteq} &: \varnothing \subseteq \Gamma \end{aligned}$$

In this formalisation, inclusion is the method by which we express weakening:

$$\mathsf{weak}_{\subseteq} : \Gamma \subseteq \Gamma\,, A$$

Furthermore, stack membership is monotonic with respect to inclusion:

$$\mathsf{mono}_{\in} \;:\; \Gamma \subseteq \Gamma' \to A \in \Gamma \to A \in \Gamma'$$

Inclusion on stack pairs is defined as a pair of inclusions, $\Gamma\,;\Delta \subseteq^2 \Gamma'\,;\Delta'' \;=\; \Gamma \subseteq \Gamma' \wedge \Delta \subseteq \Delta'$. The letter ψ refers to stack pair inclusions.

It follows directly from the definition that inclusion on stack pairs is reflexive and transitive, as well:

$$\begin{aligned}
&\mathsf{refl}_{\subseteq^2} \;:\; \Gamma\,;\Delta \subseteq^2 \Gamma\,;\Delta \\
&\mathsf{trans}_{\subseteq^2} \;:\; \Gamma\,;\Delta \subseteq^2 \Gamma'\,;\Delta' \;\to\; \Gamma'\,;\Delta' \subseteq^2 \Gamma''\,;\Delta'' \;\to\; \Gamma\,;\Delta \subseteq^2 \Gamma''\,;\Delta''
\end{aligned}$$

1.3 Syntactic entailment

Proof trees, or derivations in the $\lambda^{\Box}$-calculus, are formalised as an indexed type family, $\Gamma\,;\Delta \vdash A$, after Altenkirch and Reus[4]. This ensures that only well-typed terms are representable:

$$\frac{i : A \in \Gamma}{\textbf{var}\ i : \Gamma\,;\Delta \vdash A} \qquad \frac{i : A \in \Delta}{\textbf{mvar}\ i : \Gamma\,;\Delta \vdash A}$$

$$\frac{M : \Gamma, A\,;\Delta \vdash B}{\textbf{lam}\ M : \Gamma\,;\Delta \vdash A \Rightarrow B} \qquad \frac{M : \varnothing\,;\Delta \vdash A}{\textbf{quot}\ M : \Gamma\,;\Delta \vdash \Box A}$$

$$\frac{M : \Gamma\,;\Delta \vdash A \Rightarrow B \quad N : \Gamma\,;\Delta \vdash A}{\textbf{app}\ M\ N : \Gamma\,;\Delta \vdash B} \qquad \frac{M : \Gamma\,;\Delta \vdash \Box A \quad N : \Gamma\,;\Delta, A \vdash C}{\textbf{eval}\ M\ N : \Gamma\,;\Delta \vdash C}$$

Multiple terms in a single context can be represented as a pointwise lifting of syntactic entailment to stacks of types, $\Gamma\,;\Delta \vdash_\star \Xi$, constituting a *syntactic environment*. The letters γ, δ, ξ refer to syntactic environments.

Note that the **quot** form wraps a closed derivation, $\varnothing\,;\Delta \vdash A$, or a proof tree in which only modal assumptions are made. We view the collection of modal assumptions, Δ, as a typed specification of modal holes left in a given derivation. These holes must be plugged in order for a derivation to fully evaluate. Our evaluation function requires us to supply a syntactic environment of *modal evidence,* containing a closed derivation for each modal assumption.

Syntactic environments are both non-modally and modally reflexive:

$$\begin{aligned}
&\mathsf{refl}_{\vdash_\star} \;:\; \Gamma\,;\Delta \vdash_\star \Gamma \\
&\mathsf{mrefl}_{\vdash_\star} \;:\; \Gamma\,;\Delta \vdash_\star \Delta
\end{aligned}$$

Furthermore, syntactic entailment is monotonic with respect to stack pair inclusion:

$$\begin{aligned}
&\mathsf{mono}_{\vdash} \;:\; \Gamma\,;\Delta \subseteq^2 \Gamma'\,;\Delta' \;\to\; \Gamma\,;\Delta \vdash A \;\to\; \Gamma'\,;\Delta' \vdash A \\
&\mathsf{mono}_{\vdash_\star} \;:\; \Gamma\,;\Delta \subseteq^2 \Gamma'\,;\Delta' \;\to\; \Gamma\,;\Delta \vdash_\star \Xi \;\to\; \Gamma'\,;\Delta' \vdash_\star \Xi
\end{aligned}$$

The proof of monotonicity of syntactic entailment, $\mathsf{mono}_{\vdash}$, proceeds by induction, using monotonicity of stack membership, $\mathsf{mono}_{\in}$, in the **var** and **mvar** case.

Our soundness proof requires the ability to *graft proof trees,* which we view as a form of simultaneous substitution, or a variant of the cut rule. Grafting is the method by which we plug holes left in a given

derivation, effectively performing call-by-name (CBN):

$$
\begin{aligned}
&\mathsf{graft}_{\in} \;:\; \Gamma\,;\Delta \vdash_{\star} \Xi \;\to\; A \in \Xi \;\to\; \Gamma\,;\Delta \vdash A \\
&\mathsf{graft}_{\in}\ (\xi\ ,\ M)\ \mathbf{top} \;=\; M \\
&\mathsf{graft}_{\in}\ (\xi\ ,\ N)\ (\mathbf{pop}\ i) \;=\; \mathsf{graft}_{\in}\ \xi\ i
\end{aligned}
$$

$$
\begin{aligned}
&\mathsf{graft}_{\vdash} \;:\; \Gamma\,;\Delta \vdash_{\star} \Gamma' \;\to\; \varnothing\,;\Delta \vdash_{\star} \Delta' \;\to\; \Gamma'\,;\Delta' \vdash C \;\to\; \Gamma\,;\Delta \vdash C \\
&\mathsf{graft}_{\vdash}\ \gamma\ \delta\ (\mathbf{var}\ i) \;=\; \mathsf{graft}_{\in}\ \gamma\ i \\
&\mathsf{graft}_{\vdash}\ \gamma\ \delta\ (\mathbf{mvar}\ i) \;=\; \mathsf{mono}_{\vdash}\ (\mathsf{bot}_{\subseteq}, \mathsf{refl}_{\subseteq})\ (\mathsf{graft}_{\in}\ \delta\ i) \\
&\mathsf{graft}_{\vdash}\ \gamma\ \delta\ (\mathbf{lam}\ M) \;=\; \mathbf{lam}\ (\mathsf{graft}_{\vdash}\ (\mathsf{mono}_{\vdash_{\star}}\ (\mathsf{weak}_{\subseteq}, \mathsf{refl}_{\subseteq})\ \gamma\ ,\ (\mathbf{var}\ \mathbf{top}))\ \delta\ M) \\
&\mathsf{graft}_{\vdash}\ \gamma\ \delta\ (\mathbf{app}\ M\ N) \;=\; \mathbf{app}\ (\mathsf{graft}_{\vdash}\ \gamma\ \delta\ M)\ (\mathsf{graft}_{\vdash}\ \gamma\ \delta\ N) \\
&\mathsf{graft}_{\vdash}\ \gamma\ \delta\ (\mathbf{quot}\ M) \;=\; \mathbf{quot}\ (\mathsf{graft}_{\vdash}\ \varnothing\ \delta\ M) \\
&\mathsf{graft}_{\vdash}\ \gamma\ \delta\ (\mathbf{eval}\ M\ N) \;=\; \mathbf{eval}\ (\mathsf{graft}_{\vdash}\ \gamma\ \delta\ M) \\
&\qquad\qquad (\mathsf{graft}_{\vdash}\ (\mathsf{mono}_{\vdash_{\star}}\ (\mathsf{refl}_{\subseteq}, \mathsf{weak}_{\subseteq})\ \gamma) \\
&\qquad\qquad\quad (\mathsf{mono}_{\vdash_{\star}}\ (\mathsf{refl}_{\subseteq}, \mathsf{weak}_{\subseteq})\ \delta\ ,\ (\mathbf{mvar}\ \mathbf{top}))\ N)
\end{aligned}
$$

1.4 Normal forms

We define proof trees in normal form, $\Gamma\,;\Delta \vdash_{\mathsf{nf}} A$, mutually with proof trees in *neutral form,* $\Gamma\,;\Delta \vdash_{\mathsf{ne}} A$. Neutral forms correspond to terms that are blocked from evaluation by the presence of an assumption, as described by Altenkirch[3]:

$$\frac{M : \Gamma, A\,;\Delta \vdash_{\mathsf{nf}} B}{\mathbf{lam}_{\mathsf{nf}}\ M : \Gamma\,;\Delta \vdash_{\mathsf{nf}} A \Rightarrow B} \qquad \frac{i : A \in \Gamma}{\mathbf{var}_{\mathsf{ne}}\ i : \Gamma\,;\Delta \vdash_{\mathsf{ne}} A}$$

$$\frac{M : \varnothing\,;\Delta \vdash A}{\mathbf{quot}_{\mathsf{nf}}\ M : \Gamma\,;\Delta \vdash_{\mathsf{nf}} \Box A} \qquad \frac{i : A \in \Delta}{\mathbf{mvar}_{\mathsf{ne}}\ i : \Gamma\,;\Delta \vdash_{\mathsf{ne}} A}$$

$$\frac{M : \Gamma\,;\Delta \vdash_{\mathsf{ne}} A \Rightarrow B \quad N : \Gamma\,;\Delta \vdash_{\mathsf{nf}} A}{\mathbf{app}_{\mathsf{ne}}\ M\ N : \Gamma\,;\Delta \vdash_{\mathsf{ne}} B}$$

$$\frac{M : \Gamma\,;\Delta \vdash_{\mathsf{ne}} A}{\mathbf{ne}_{\mathsf{nf}}\ M : \Gamma\,;\Delta \vdash_{\mathsf{nf}} A} \qquad \frac{M : \Gamma\,;\Delta \vdash_{\mathsf{ne}} \Box A \quad N : \Gamma\,;\Delta, A \vdash_{\mathsf{nf}} C}{\mathbf{eval}_{\mathsf{ne}}\ M\ N : \Gamma\,;\Delta \vdash_{\mathsf{ne}} C}$$

Syntactic entailment in normal and neutral form is monotonic with respect to stack pair inclusion:

$$
\begin{aligned}
&\mathsf{mono}_{\vdash_{\mathsf{nf}}} \;:\; \Gamma\,;\Delta \subseteq^2 \Gamma'\,;\Delta' \;\to\; \Gamma\,;\Delta \vdash_{\mathsf{nf}} A \;\to\; \Gamma'\,;\Delta' \vdash_{\mathsf{nf}} A \\
&\mathsf{mono}_{\vdash_{\mathsf{ne}}} \;:\; \Gamma\,;\Delta \subseteq^2 \Gamma'\,;\Delta' \;\to\; \Gamma\,;\Delta \vdash_{\mathsf{ne}} A \;\to\; \Gamma'\,;\Delta' \vdash_{\mathsf{ne}} A
\end{aligned}
$$

It is important to note that the $\mathbf{quot}_{\mathsf{nf}}$ normal form wraps a closed, non-normal form derivation, $\varnothing\,;\Delta \vdash A$, for we do not wish to evaluate under **quot**. This matches the definition of convertibility for the $\lambda^{\Box}$-calculus given by Pfenning and Davies, which is not congruent under the $\Box$ introduction rule.

Some say that in LISP, code is data. We prefer to say that code may be **quot**ed to become data, and such data may be **eval**uated to become code.

2 Semantics

2.1 Model

An *introspective Kripke model* is a tuple containing a *set of possible worlds,* World, a *family of ground sets,* $\mathsf{G}\,w$, an *intuitionistic accessibility relation*, $w \leq w'$, a *modal accessibility relation*, $w\ \mathsf{R}\ w'$, and two novel conditions, described below.

The family of ground sets is intended to provide a semantics for the base type, $\bullet$. As usual in an intuitionistic Kripke model, we require this family to be monotonic with respect to intuitionistic accessibility, $\mathsf{mono_G}\ \psi\ g$. Both accessibility relations must be reflexive and transitive, as well.

Our models include a *vindication condition*, $\leq\!\!\rightarrow\mathsf{R}\ \psi$, which states that intuitionistic accessibility must imply modal accessibility. Vindication is the only method by which the two relations interact in our models.

Additionally, we introduce an *introspection condition,* $\mathsf{peek}\ w$. This unusual condition allows us to intensionally inspect a given world, obtaining the assumptions that have been made in that world:

$$
\begin{aligned}
\mathsf{mono_G} &: w \leq w' \to \mathsf{G}\,w \to \mathsf{G}\,w' \\
\mathsf{refl}_{\leq} &: w \leq w \\
\mathsf{trans}_{\leq} &: w \leq w' \to w' \leq w'' \to w \leq w'' \\
\mathsf{refl_R} &: w\ \mathsf{R}\ w \\
\mathsf{trans_R} &: w\ \mathsf{R}\ w' \to w'\ \mathsf{R}\ w'' \to w\ \mathsf{R}\ w'' \\
\leq\!\!\rightarrow\mathsf{R} &: w \leq w' \to w\ \mathsf{R}\ w' \\
\mathsf{peek} &: \mathsf{World} \to \mathsf{Context}
\end{aligned}
$$

Since a context is defined as a pair of type stacks, $\pi_2\ (\mathsf{peek}\ w)$ yields a stack of modal assumptions, Δ. The ability to obtain only the modal assumptions made at a given world is precisely what enables us to embed a closed derivation, $\varnothing\,;\Delta \vdash A$, in the semantics.

2.2 Forcing

We define a *strong forcing relation*, $w \Vdash A$, which explains what it means for the proposition A to be true in the world w. This relation is defined mutually with *non-strong forcing,* $w \Vdash\!\!\!\vdash A$, in a continuation-passing style similar to Ilik's. The difference is that we do not require *exploding worlds* in our models for this definition, thanks to our introspection condition.

A *value environment*, $w \Vdash_\star \Xi$, is obtained by a pointwise lifting of strong forcing to stacks of types. The letters ρ, σ refer to value environments.

$$
\begin{aligned}
w \Vdash A &= w \leq w' \to (w' \leq w'' \to w'' \Vdash\!\!\!\vdash A \to \mathsf{peek}\ w'' \vdash_{\mathsf{nf}} C) \to \mathsf{peek}\ w' \vdash_{\mathsf{nf}} C \\
w \Vdash\!\!\!\vdash \bullet &= \mathsf{G}\,w \\
w \Vdash\!\!\!\vdash A \Rightarrow B &= w \leq w' \to w' \Vdash A \to w' \Vdash B \\
w \Vdash\!\!\!\vdash \Box A &= w\ \mathsf{R}\ w' \to \varnothing\,;\pi_2\ (\mathsf{peek}\ w') \vdash A \wedge w' \Vdash A
\end{aligned}
$$

We are now justified in saying that $\Box A$ is true in the model if and only if there strongly exists a term that corresponds to a closed, non-normal form derivation of A, and A is true in the model. To rephrase: we can say that $\Box A$ is true as long as we have *evidence* for the fact that A is true.

3 Soundness

3.1 Semantic entailment

We define a *semantic entailment relation*, $\Gamma\,;\Delta \vDash A$, as strong forcing in all worlds of all models:

$$\begin{aligned}\Gamma\,;\Delta \vDash A \;=\; & w \Vdash_\star \Gamma \;\to \\ & (w\ \mathsf{R}\ w' \to \varnothing\,;\pi_2\ (\mathsf{peek}\, w') \vdash_\star \Delta) \;\to \\ & (w\ \mathsf{R}\ w' \to w' \Vdash_\star \Delta) \;\to \\ & w \Vdash A\end{aligned}$$

The above definition can also be understood as strong forcing in a three-part environment, composed of:

1. ρ, a non-modal value environment,
2. δ, a syntactic environment of *modal evidence,* containing a closed, non-normal form derivation for each modal assumption, and
3. σ, a modal value environment.

3.2 Reflection

We demonstrate the soundness of our models with respect to the $\lambda^{\Box}$-calculus by defining an *evaluation function,* $\mathsf{reflect}\ M$, which maps derivations to semantic objects, or values within the model. The soundness proof is written in a continuation-passing style, after Ilik, with the continuation monad defined as follows:

$$\begin{aligned}&\mathsf{return}\ :\ w \Vdash\!\Vdash A \;\to\; w \Vdash A \\ &\mathsf{return}\ a \;=\; \lambda\,\psi\,\kappa\,. \\ &\qquad\qquad \kappa\ \mathsf{refl}_{\leq}\ (\mathsf{mono}_{\Vdash\!\Vdash}\ \psi\ a)\end{aligned}$$

$$\begin{aligned}&\mathsf{bind}\ :\ w \Vdash A \;\to\; (w \leq w' \to w' \Vdash\!\Vdash A \to w' \Vdash B) \;\to\; w \Vdash B \\ &\mathsf{bind}\ a\ \kappa \;=\; \lambda\,\psi\,\kappa'\,. \\ &\qquad\qquad a\ \psi\ (\lambda\,\psi'\,a'\,. \\ &\qquad\qquad\quad \kappa\ (\mathsf{trans}_{\leq}\ \psi\ \psi')\ a'\ \mathsf{refl}_{\leq}\ (\lambda\,\psi''\,a''\,. \\ &\qquad\qquad\qquad \kappa'\ (\mathsf{trans}_{\leq}\ \psi'\ \psi'')\ a''))\end{aligned}$$

The ability to look up values in value environments, $\mathsf{lookup}\ i\ \xi$, is required in the **var** and **mvar** case of the soundness proof:

$$\begin{aligned}&\mathsf{lookup}\ :\ A \in \Xi \;\to\; w \Vdash_\star \Xi \;\to\; w \Vdash A \\ &\mathsf{lookup}\ \mathbf{top}\qquad (\xi\ ,\ s) \;=\; s \\ &\mathsf{lookup}\ (\mathbf{pop}\ i)\ (\xi\ ,\ t) \;=\; \mathsf{lookup}\ i\ \xi\end{aligned}$$

The proof of soundness makes use of the vindication condition, $\leq\!\!\rightarrow\mathsf{R}$, for the purpose of transporting modal evidence, δ, and modal values, σ, between worlds:

$$
\begin{aligned}
&\mathsf{reflect}\ :\ \Gamma\,;\Delta \vdash A \ \rightarrow\ \Gamma\,;\Delta \vDash A\\
&\mathsf{reflect}\ (\mathbf{var}\ i) \quad \rho\ \delta\ \sigma\ =\ \mathsf{lookup}\ i\ \rho\\
&\mathsf{reflect}\ (\mathbf{mvar}\ i) \quad \rho\ \delta\ \sigma\ =\ \mathsf{lookup}\ i\ (\sigma\ \mathsf{refl}_{\mathsf{R}})\\
&\mathsf{reflect}\ (\mathbf{lam}\ M) \quad \rho\ \delta\ \sigma\ =\ \mathsf{return}\ (\lambda\ \psi\ a\ .\\
&\qquad\qquad \mathsf{reflect}\ M\ (\mathsf{mono}_{\Vdash_\star}\ \psi\ \rho\ ,\ a)\\
&\qquad\qquad\qquad (\lambda\ \mu\ .\ \delta\ (\mathsf{trans}_{\mathsf{R}}\ (\leq\!\!\rightarrow\mathsf{R}\ \psi)\ \mu))\\
&\qquad\qquad\qquad (\lambda\ \mu\ .\ \sigma\,(\mathsf{trans}_{\mathsf{R}}\ (\leq\!\!\rightarrow\mathsf{R}\ \psi)\ \mu)))\\
&\mathsf{reflect}\ (\mathbf{app}\ M\ N)\ \rho\ \delta\ \sigma\ =\ \mathsf{bind}\ (\mathsf{reflect}\ M\ \rho\ \delta\ \sigma)\ (\lambda\ \psi\ f\ .\\
&\qquad\qquad f\ \mathsf{refl}_{\leq}\ (\mathsf{mono}_{\Vdash}\ \psi\ (\mathsf{reflect}\ N\ \rho\ \delta\ \sigma)))\\
&\mathsf{reflect}\ (\mathbf{quot}\ M) \quad \rho\ \delta\ \sigma\ =\ \mathsf{return}\ (\lambda\ \mu\ .\\
&\qquad\qquad \mathsf{graft}_{\vdash}\ \varnothing\ (\delta\ \mu)\ M\ ,\\
&\qquad\qquad \mathsf{reflect}\ M\ \varnothing\\
&\qquad\qquad\qquad (\lambda\ \mu'\ .\ \delta\ (\mathsf{trans}_{\mathsf{R}}\ \mu\ \mu'))\\
&\qquad\qquad\qquad (\lambda\ \mu'\ .\ \sigma\,(\mathsf{trans}_{\mathsf{R}}\ \mu\ \mu')))\\
&\mathsf{reflect}\ (\mathbf{eval}\ M\ N)\ \rho\ \delta\ \sigma\ =\ \mathsf{bind}\ (\mathsf{reflect}\ M\ \rho\ \delta\ \sigma)\ (\lambda\ \psi\ f\ .\\
&\qquad\qquad \mathsf{reflect}\ M\ (\mathsf{mono}_{\Vdash_\star}\ \psi\ \rho)\\
&\qquad\qquad\qquad (\lambda\ \mu\ .\ \delta\ (\mathsf{trans}_{\mathsf{R}}\ (\leq\!\!\rightarrow\mathsf{R}\ \psi)\ \mu)\ ,\ \pi_1\ (f\ \mu))\\
&\qquad\qquad\qquad (\lambda\ \mu\ .\ \sigma\,(\mathsf{trans}_{\mathsf{R}}\ (\leq\!\!\rightarrow\mathsf{R}\ \psi)\ \mu)\ ,\ \pi_2\ (f\ \mu)))
\end{aligned}
$$

To understand the **quot** case, let us recall the meaning of $\Box A$, namely, a pair that contains a closed, non-normal form derivation of A, and a value of type A. The first element of the returned pair is the result of grafting all necessary modal evidence from the syntactic environment δ on top of M, or plugging modal holes left in M. The second element of the result is a value that corresponds to the first element.

4 Completeness

4.1 Canonical model

A *canonical model* is a model that allows each value to be reified as a normal form. We begin the construction of a canonical introspective Kripke model by identifying worlds with contexts. It immediately follows that the introspection condition has been the identity function all along.

The family of ground sets can be defined as neutral form derivations that correspond to terms of the base type, $\mathsf{G}\,w\ =\ w \vdash_{\mathsf{ne}} \bullet$. This preserves the monotonicity condition.

Since a context is defined as a pair of stacks, intuitionistic accessibility becomes stack pair inclusion, $w \leq w'\ =\ w \subseteq^2 w'$, and modal accessibility becomes inclusion on the second element of each context, $w\ \mathsf{R}\ w'\ =\ \pi_2\ w \subseteq \pi_2\ w'$. The reflexivity and transitivity conditions are preserved.

Inclusion on stack pairs is defined as a pair of inclusions. Therefore, the vindication condition, $\leq\!\!\rightarrow\mathsf{R}$, can be identified with the second projection function, π_2.

We are now ready to define a function that maps proof trees to values within the canonical model, $\mathsf{reflect}_{\mathsf{c}}\ M$, mutually with an inverse mapping, $\mathsf{reify}_{\mathsf{c}}\ \kappa$, which reads back the resulting values as derivations

in normal form. Both definitions proceed by induction on the type of the derivation.

$$
\begin{array}{lll}
\mathsf{reflect}_{\mathsf{c}} & : & \Gamma\,;\Delta \vdash_{\mathsf{ne}} A \;\to\; \Gamma\,;\Delta \Vdash A \\
\mathsf{reflect}_{\mathsf{c}}\ \{\bullet\} & M & = \mathsf{return}\ M \\
\mathsf{reflect}_{\mathsf{c}}\ \{A \Rightarrow B\}\ M & = & \mathsf{return}\ (\lambda\,\psi\ a\ .\ \\
 & & \quad \mathsf{reflect}_{\mathsf{c}}\ (\mathbf{app}_{\mathsf{ne}}\ (\mathsf{mono}_{\vdash_{\mathsf{ne}}}\ \psi\ M)\ (\mathsf{reify}_{\mathsf{c}}\ a))) \\
\mathsf{reflect}_{\mathsf{c}}\ \{\Box A\} & M & = \lambda\,\psi\ \kappa\ . \\
 & & \quad \mathbf{ne}_{\mathsf{nf}}\ (\mathbf{eval}_{\mathsf{ne}}\ (\mathsf{mono}_{\vdash_{\mathsf{ne}}}\ \psi\ M)\ (\kappa\ (\mathsf{refl}_{\subseteq}, \mathsf{weak}_{\subseteq})\ (\lambda\,\mu'\ . \\
 & & \qquad \mathsf{mono}_{\vdash}\ (\mathbf{done}, \mu')\ (\mathbf{mvar}\ \ \mathbf{top}), \\
 & & \qquad \mathsf{reflect}_{\mathsf{c}}\ (\mathsf{mono}_{\vdash_{\mathsf{ne}}}\ (\mathsf{bot}_{\subseteq}, \mu')\ (\mathbf{mvar}_{\mathsf{ne}}\ \ \mathbf{top})))))
\end{array}
$$

$$
\begin{array}{lll}
\mathsf{reify}_{\mathsf{c}} & : & \Gamma\,;\Delta \Vdash A \;\to\; \Gamma\,;\Delta \vdash_{\mathsf{nf}} A \\
\mathsf{reify}_{\mathsf{c}}\ \{\bullet\} & \kappa & = \kappa\ \mathsf{refl}_{\subseteq^2}\ (\lambda\,\psi\ o\ . \\
 & & \quad \mathbf{ne}_{\mathsf{nf}}\ o) \\
\mathsf{reify}_{\mathsf{c}}\ \{A \Rightarrow B\} & \kappa & = \kappa\ \mathsf{refl}_{\subseteq^2}\ (\lambda\,\psi\ f\ . \\
 & & \quad \mathbf{lam}_{\mathsf{nf}}\ (\mathsf{reify}_{\mathsf{c}}\ (f\ (\mathsf{weak}_{\subseteq}, \mathsf{refl}_{\subseteq})\ (\mathsf{reflect}_{\mathsf{c}}\ (\mathbf{var}_{\mathsf{ne}}\ \ \mathbf{top})))))) \\
\mathsf{reify}_{\mathsf{c}}\ \{\Box A\} & \kappa & = \kappa\ \mathsf{refl}_{\subseteq^2}\ (\lambda\,\psi\ f\ . \\
 & & \quad \mathbf{quot}_{\mathsf{nf}}\ (\pi_1\ (f\ \ \mathsf{refl}_{\subseteq})))
\end{array}
$$

By reflecting non-modal and modal assumptions into the canonical model, we obtain that value environments are both non-modally and modally reflexive:

$$
\begin{array}{ll}
\mathsf{refl}_{\Vdash_\star} & : \Gamma\,;\Delta \Vdash_\star \Gamma \\
\mathsf{mrefl}_{\Vdash_\star} & : \Gamma\,;\Delta \Vdash_\star \Delta
\end{array}
$$

4.2 Reification

Defining an *inverse of the evaluation function,* $\mathsf{reify}\ o$, demonstrates the completeness of our models with respect to the $\lambda^{\Box}$-calculus. The proof proceeds by reading back values from the canonical model, in a three-part environment of reflected assumptions:

$$
\begin{array}{lll}
\mathsf{reify} & : & \Gamma\,;\Delta \vDash A \;\to\; \Gamma\,;\Delta \vdash_{\mathsf{nf}} A \\
\mathsf{reify}\ o & = & \mathsf{reify}_{\mathsf{c}}\ (o\ \mathsf{refl}_{\Vdash_\star} \\
 & & \qquad (\lambda\,\mu\ .\ \mathsf{mono}_{\vdash_\star}\ (\mathsf{refl}_{\subseteq}, \mu)\ \mathsf{mrefl}_{\vdash_\star}) \\
 & & \qquad (\lambda\,\mu\ .\ \mathsf{mono}_{\Vdash_\star}\ (\mathsf{refl}_{\subseteq}, \mu)\ \mathsf{mrefl}_{\Vdash_\star})
\end{array}
$$

4.3 Normalisation

Finally, we obtain a normalisation procedure as composition of the soundness and completeness proofs:

$$
\begin{array}{lll}
\mathsf{nbe} & : & \Gamma\,;\Delta \vdash A \;\to\; \Gamma\,;\Delta \vdash_{\mathsf{nf}} A \\
\mathsf{nbe} & = & \mathsf{reify} \circ \mathsf{reflect}
\end{array}
$$

Conclusion

We have presented a class of introspective Kripke models that is sound and complete with respect to the $\lambda^{\Box}$-calculus of Pfenning and Davies[28]. As far as we know, our work supplies the first constructive proof of completeness for an S4-style modal logic. Our next steps will involve verifying the correctness of the NbE procedure against a suitable convertibility relation.

We hope that the present work clears a path towards a computational interpretation of Artemov's[5] logic of proofs (LP). Some of our results may also be applicable to the contextual modal type theory of Nanevski *et al.*[25].

Acknowledgements

The author is deeply grateful to Andreas Abel, Guillaume Allais, Ahmad Salim Al-Sibahi, Roy Dyckhoff, Michael Gabbay, Paolo G. Giarrusso, Tom Jack, Roman Kireev, Darryl McAdams, Conor McBride, Dominic Orchard, Ida Szubert, Andrea Vezzosi, and Tomasz Wierzbicki, for many fruitful discussions over the years.

References

[1] A. Abel (2013): *Normalization by evaluation: Dependent types and impredicativity*. Habilitation thesis, Ludwig-Maximilians-Universität München. Available at http://tcs.ifi.lmu.de/~abel/habil.pdf. 1

[2] N. Alechina, M. Mendler, V. de Paiva & E. Ritter (2001): *Categorical and Kripke semantics for constructive S4 modal logic*. In: *Computer Science Logic (CSL'01)*, *LNCS* 2142, pp. 292–307, doi:10.1007/3-540-44802-0_21. 1

[3] T. Altenkirch (1993): *Proving strong normalization of CC by modifying realizability semantics*. In: *Types for Proofs and Programs (TYPES'93)*, *LNCS* 806, pp. 3–18, doi:10.1007/3-540-58085-9_70. 6

[4] T. Altenkirch & B. Reus (1999): *Monadic presentations of lambda terms using generalized inductive types*. In: *Computer Science Logic (CSL'99)*, *LNCS* 1683, pp. 453–468, doi:10.1007/3-540-48168-0_32. 5

[5] S. N. Artemov (2001): *Explicit provability and constructive semantics*. *Bulletin of Symbolic Logic* 7(1), pp. 1–36, doi:10.2307/2687821. 3, 11

[6] U. Berger & H. Schwichtenberg (1991): *An inverse of the evaluation functional for typed lambda-calculus*. In: *Logic in Computer Science (LICS'91)*, pp. 203–211, doi:10.1109/LICS.1991.151645. 1

[7] G. M. Bierman & V. de Paiva (2000): *On an intuitionistic modal logic*. *Studia Logica* 65(3), pp. 383–416, doi:0.1023/A:1005291931660. 1

[8] M. Božić & K. Došen (1984): *Models for normal intuitionistic modal logics*. *Studia Logica* 43(3), pp. 217–245, doi:10.1007/BF02429840. 1

[9] J. M. Chapman (2009): *Type checking and normalisation*. Ph.D. thesis, University of Nottingham. Available at http://jmchapman.github.io/papers/thesis.pdf. 4

[10] C. Coquand (1993): *From semantics to rules: A machine assisted analysis*. In: *Computer Science Logic (CSL'93)*, *LNCS* 832, pp. 91–105, doi:10.1007/BFb0049326. 1

[11] T. Coquand & P. Dybjer (1997): *Intuitionistic model constructions and normalization proofs*. *Mathematical Structures in Computer Science* 7(1), pp. 75–94, doi:10.1017/S0960129596002150. 3

[12] O. Danvy (1996): *Type-directed partial evaluation*. In: *Principles of Programming Languages (POPL'96)*, pp. 242–257, doi:10.1145/237721.237784. 2

[13] R. Dyckhoff (2016): *Some remarks on proof-theoretic semantics*. In: *Advances in Proof-Theoretic Semantics*, *Trends in Logic* 43, Springer, pp. 79–93, doi:10.1007/978-3-319-22686-6_5. 2

[14] W. B. Ewald (1986): *Intuitionistic tense and modal logic*. Journal of Symbolic Logic 51(1), pp. 166–179, doi:10.2307/2273953. 1

[15] G. Fischer Servi (1984): *Axiomatizations for some intuitionistic modal logics*. Rendiconti del Seminario Matematico Università Politecnico di Torino 42, pp. 179–194. Available at http://seminariomatematico.unito.it/rendiconti/cartaceo/. Soon online. 1

[16] M. J. Gabbay & A. Nanevski (2013): *Denotation of contextual modal type theory: Syntax and meta-programming*. Journal of Applied Logic 11(1), pp. 1–29, doi:10.1016/j.jal.2012.07.002. 3

[17] J.-Y. Girard, P. Taylor & Y. Lafont (1989): *Proofs and Types*. Cambridge University Press. Available at http://paultaylor.eu/stable/prot.pdf. 2

[18] K. Gödel (1933/1986): *Eine Interpretation des intuitionistischen Aussagenkalküls*. In: *Kurt Gödel: Collected Works, Vol. I: Publications 1929–1936*, Oxford University Press, pp. 300–303. Available at https://global.oup.com/academic/product/collected-works-9780195039641. 3

[19] R. Iemhoff (2001): *Provability logic and admissible rules*. Ph.D. thesis, University of Amsterdam. Available at http://phil.uu.nl/~iemhoff/Mijn/Papers/proeve.pdf. 2

[20] D. Ilik (2013): *Continuation-passing style models complete for intuitionistic logic*. Annals of Pure and Applied Logic 164(6), pp. 651–662, doi:10.1016/j.apal.2012.05.003. 2

[21] S. A. Kripke (1965): *Semantical analysis of intuitionistic logic I*. Studies in Logic and the Foundations of Mathematics 40, pp. 92–130, doi:10.1016/S0049-237X(08)71685-9. 2

[22] P. Martin-Löf (1975): *An intuitionistic theory of types: Predicative part*. Studies in Logic and the Foundations of Mathematics 80, pp. 73–118, doi:10.1016/S0049-237X(08)71945-1. 1

[23] J. McCarthy, P. W. Abrahams, D. J. Edwards, T. P. Hart & M. I. Levin (1962): *LISP 1.5 Programmer's Manual*. MIT Press. Available at http://softwarepreservation.org/projects/LISP/book/LISP%201.5%20Programmers%20Manual.pdf. 1

[24] J. C. C. McKinsey & A. Tarski (1948): *Some theorems about the sentential calculi of Lewis and Heyting*. Journal of Symbolic Logic 13(1), pp. 1–15, doi:10.2307/2268135. 3

[25] A. Nanevski, F. Pfenning & B. Pientka (2008): *Contextual modal type theory*. ACM Transactions on Computational Logic 9(3), pp. 23:1–23:49, doi:10.1145/1352582.1352591. 3, 11

[26] U. Norell (2007): *Towards a practical programming language based on dependent type theory*. Ph.D. thesis, Chalmers University of Technology. Available at http://www.cse.chalmers.se/~ulfn/papers/thesis.pdf. 3

[27] H. Ono (1977): *On some intuitionistic modal logics*. Publications of the Research Institute for Mathematical Sciences 13, pp. 687–722, doi:10.2977/prims/1195189604. 1

[28] F. Pfenning & R. Davies (2001): *A judgmental reconstruction of modal logic*. Mathematical Structures in Computer Science 11(4), pp. 511–540, doi:10.1017/S0960129501003322. 1, 11

[29] G. D. Plotkin & C. Stirling (1986): *A framework for intuitionistic modal logics*. In: *Theoretical Aspects of Reasoning about Knowledge (TARK'86)*, pp. 399–406. Available at http://homepages.inf.ed.ac.uk/gdp/publications/Framework_Int_Modal_Logics.pdf. 1

[30] T. Sheard (2001): *Accomplishments and research challenges in meta-programming*. In: *Semantics, Applications, and Implementation of Program Generation (SAIG'01)*, *LNCS* 2196, Springer, pp. 2–44, doi:10.1007/3-540-44806-3_2. 1

[31] A. Simpson (1994): *The proof theory and semantics of intuitionistic modal logic*. Ph.D. thesis, University of Edinburgh. Available at http://homepages.inf.ed.ac.uk/als/Research/thesis.pdf. 1

[32] D. A. Turner (2004): *Total functional programming*. Journal of Universal Computer Science 10(7), pp. 751–768, doi:10.3217/jucs-010-07-0751. 1

[33] P. Wadler (2015): *Propositions as types*. Communications of the ACM 58(12), pp. 75–84, doi:10.1145/2699407. 1

[34] D. Wijesekera (1990): *Constructive modal logics I*. Annals of Pure and Applied Logic 50(3), pp. 271–301, doi:10.1016/0168-0072(90)90059-B. 1