Continuation-passing Style Models Complete for Intuitionistic Logic

Danko Ilik[1]

Ecole Polytechnique, INRIA, CNRS & Université Paris Diderot
Address: INRIA PI-R2, 23 avenue d'Italie, CS 81321, 75214 Paris Cedex 13, France
E-mail: danko.ilik@polytechnique.edu

Abstract

A class of models is presented, in the form of continuation monads polymorphic for first-order individuals, that is sound and complete for minimal intuitionistic predicate logic (including disjunction and the existential quantifier). The proofs of soundness and completeness are constructive and the computational content of their composition is, in particular, a β-normalisation-by-evaluation program for simply typed lambda calculus with sum types. Although the inspiration comes from Danvy's type-directed partial evaluator for the same lambda calculus, the use of delimited control operators (i.e. computational effects) is avoided. The role of polymorphism is crucial – dropping it allows one to obtain a notion of model complete for classical predicate logic.

Keywords: intuitionistic logic, completeness, Kripke models, Double-negation Shift, normalisation by evaluation
2000 MSC: 03B20, 03B35, 03B40, 68N18, 03F55, 03F50, 03B55

1. Introduction

Although Kripke models are standard semantics for intuitionistic logic, there is as yet no (simple) constructive proof of their completeness when one considers all the logical connectives. While Kripke's original proof [25] was classical, Veldman gave an intuitionistic one [32] by using Brouwer's Fan Theorem to handle disjunction and the existential quantifier. To see what the computational content behind Veldman's proof is, one might consider a realisability interpretation of the Fan Theorem (for example [5]), but, all known realisers being defined by general recursion, due to the absence of an elementary proof of their termination, it is not clear whether one can think of the program using them as a constructive proof or not.

[1]Present address: Faculty of Informatics, University "Goce Delčev", PO Box 201, 2000 Štip, Macedonia; E-mail: dankoilik@gmail.com

On the other hand, a connection between normalisation-by-evaluation (NBE) [6] for simply typed lambda calculus, $\lambda^{\rightarrow}$, and completeness of minimal intuitionistic logic for Kripke models for the fragment $\{\wedge, \Rightarrow, \forall\}$ has been made [8, 19]. We review this connection in Section 2. There, we also revisit Danvy's extension [10] of NBE from $\lambda^{\rightarrow}$ to $\lambda^{\rightarrow\vee}$, simply typed lambda calculus with sum types. Even though Danvy's algorithm is simple and elegant, he uses the full power of delimited control operators which do not yet have a typing system that permits to understand that use logically. We deal with that problem in Section 3, by modifying the notion of Kripke model so that we can give a proof of completeness for full intuitionistic logic in continuation-passing style, that is, without relying on having delimited control operators in our meta-language. In Section 4, we extract the algorithm behind the given completeness proof, a β-NBE algorithm for $\lambda^{\rightarrow\vee}$. In Section 5, we stress the importance of our models being parametric, by comparing them to similar models that are complete for classical logic [23]. We conclude with Section 6, where we also mention related work.

The proofs of Section 3 have been formalised in the Coq proof assistant in [20], which also represents an implementation of the NBE algorithm of Section 4.

2. Normalisation-by-Evaluation as Completeness

In [6], Berger and Schwichtenberg presented a proof of normalisation of $\lambda^{\rightarrow}$ which does not involve reasoning about the associated reduction relation. Instead, they interpreted λ-terms in a domain, or ambient meta-language, using an evaluation function,

$$[\![-]\!] : \Lambda \rightarrow D,$$

and then they defined an inverse to this function, which from the denotation in D directly extracts a term in $\beta\eta$-long normal form. The inverse function $\downarrow$, called *reification*, is defined by recursion on the type τ of the term, at the same time with an auxiliary function $\uparrow$, called *reflection*:

$$\begin{aligned}
\downarrow^{\tau} &: D \rightarrow \Lambda\text{-nf} & \\
\downarrow^{\tau} &:= a \mapsto a & \tau\text{-atomic} \\
\downarrow^{\tau\rightarrow\sigma} &:= S \mapsto \lambda a.\downarrow^{\sigma} (S\cdot \uparrow^{\tau} a) & a\text{-fresh}
\end{aligned}$$

$$\begin{aligned}
\uparrow^{\tau} &: \Lambda\text{-ne} \rightarrow D & \\
\uparrow^{\tau} &:= a \mapsto a & \tau\text{-atomic} \\
\uparrow^{\tau\rightarrow\sigma} &:= e \mapsto S \mapsto \uparrow^{\sigma} e(\downarrow^{\tau} S) &
\end{aligned}$$

Here, S ranges over members of D, and we used $\mapsto$ and $\cdot$ for abstraction and application at the meta-level. The classes of neutral (Λ-ne) and λ-terms in normal form (Λ-nf) are given by the following inductive definition.[2]

$$\begin{aligned}
\Lambda\text{-nf} \ni r &:= \lambda a^\tau . r^\sigma \mid e^\tau && \lambda\text{-terms in normal form}\\
\Lambda\text{-ne} \ni e &:= a^\tau \mid e^{\tau\to\sigma} r^\tau && \text{neutral } \lambda\text{-terms}
\end{aligned}$$

It was a subsequent realisation of Catarina Coquand [8], that the evaluation algorithm $[\![\cdot]\!]$ is also the one underlying the Soundness Theorem for minimal intuitionistic logic (with $\Rightarrow$ as the sole logical connective) with respect to Kripke models, and that the reification algorithm $\downarrow$ is also the one underlying the corresponding Completeness Theorem.

More precisely, the following well-known statements hold and their proofs have been machine-checked [9, 19] for the logic fragment generated by the connectives $\{\Rightarrow, \wedge, \forall\}$.

Definition 2.1. A *Kripke model* is given by a preorder $(K, \le)$ of *possible worlds*, a *quantification domain* $D(w)$ for every $w \in K$, and a relation of *forcing*, $w \Vdash X$, that interprets the predicate $X(x_1, \ldots, x_n)$ in the world w by an n-ary relation on $D(w)$, such that,

$$\text{for all } w' \ge w, D(w) \subseteq D(w'), \text{ and}$$
$$\text{for } w' \ge w, d_1, \ldots, d_n \in D(w), (w \Vdash X)(d_1, \ldots, d_n) \to (w' \Vdash X)(d_1, \ldots, d_n).$$

The relation of forcing is then extended to all formulae by the following clauses, using an explicit superscript σ substitution necessary for a precise handling of quantifiers:

$$\begin{aligned}
w \Vdash^\sigma X(x_1, \ldots, x_n) &:= (w \Vdash X)(d_1, \ldots, d_n), \text{ when } \sigma = \{x_1 \mapsto d_1, \ldots, x_n \mapsto d_n\}\\
w \Vdash^\sigma A \wedge B &:= w \Vdash^\sigma A \text{ and } w \Vdash^\sigma B\\
w \Vdash^\sigma A \vee B &:= w \Vdash^\sigma A \text{ or } w \Vdash^\sigma B\\
w \Vdash^\sigma A \Rightarrow B &:= \text{ for all } w' \ge w, w' \Vdash^\sigma A \to w' \Vdash^\sigma B\\
w \Vdash^\sigma \forall x. A(x) &:= \text{ for all } w' \ge w \text{ and } d \in D(w'), w' \Vdash^{\sigma, x\mapsto d} A(x)\\
w \Vdash^\sigma \exists x. A(x) &:= \text{ for some } d \in D(w), w \Vdash^{\sigma, x\mapsto d} A(x)\\
w \Vdash^\sigma \bot &:= \text{ false}\\
w \Vdash^\sigma \top &:= \text{ true}
\end{aligned}$$

We write $w \Vdash^\sigma \Gamma$ when w forces each formula of Γ. We write $\sigma \in D(w)$ to emphasise that all interpretations of individuals from σ live in $D(w)$.

Remark 2.2. The explicit substitution environment σ maps free variables x to interpretations of individual constants $d \in D(w)$. In the forcing clauses for $\Rightarrow$ and $\forall$, there is an implicit lifting from $\sigma \in D(w)$ to $\sigma \in D(w')$.

[2]Neutral terms are the subset of normal terms that cannot be reduced on their own, whose reduction is blocked because of a free-variable appearance. Closed $\lambda^{\to}$-terms always reduce to closed terms in normal form, never to neutral terms.

Theorem 2.3 (Soundness)**.** *If* $\Gamma \vdash p : A$ *then, in any Kripke model, for any world* w *and* $\sigma \in D(w)$*, if* $w \Vdash^\sigma \Gamma$*, then* $w \Vdash^\sigma A$*.*

Proof. By induction on the height of the derivation tree. □

Theorem 2.4 (Strong Completeness by Substitution[3])**.** *There is a model* $\mathcal{U}$ *(the "universal model") such that, given a world* w *of* $\mathcal{U}$ *and* $\sigma \in D(w)$*, if* $w \Vdash^\sigma A$*, then there exists a term* p *and a derivation in normal form* $w \vdash p : A$*.*

Proof. The universal model $\mathcal{U}$ is built by setting:

- K to be the set of contexts Γ;
- "$\leq$" to be the subset relation of contexts;
- "$\Gamma \Vdash X$" to be the set of derivations in normal form $\Gamma \vdash^{\mathrm{nf}} X$, for X a closed atomic formula;
- $D(-)$ to be the constant function that for every world gives the same set of individual terms of predicate logic.

One then proves simultaneously, by induction on the complexity of A, that the two functions defined above, reify ($\downarrow$) and reflect ($\uparrow$), are correct, that is, that $\downarrow$ maps a member of $\Gamma \Vdash^\sigma A$ to a normal proof term (derivation) $\Gamma \vdash p : A$, and that $\uparrow$ maps a neutral term (derivation) $\Gamma \vdash e : A$ to a member of $\Gamma \Vdash^\sigma A$. □

We can compose theorems 2.3 and 2.4 to obtain a normalisation procedure.

Corollary 2.5 (Normalisation-by-evaluation (NBE))**.** *For every closed* $\lambda^{\to}$*-term* p*, such that* $\vdash p : \tau$*, there exists a closed term in normal form* $p' \in \Lambda$*-nf, such that* $\vdash p' : \tau$*.*

The following corollary justifies for $\mathcal{U}$ the name "universal model".

Corollary 2.6 (Completeness, usual formulation)**.** *If in any Kripke model, for any world* w *and* $\sigma \in D(w)$*,* $w \Vdash^\sigma \Gamma$ *implies* $w \Vdash^\sigma A$*, then there exists a term* p *and a derivation* $\Gamma \vdash p : A$*.*

Proof. If $w \Vdash^\sigma \Gamma \to w \Vdash^\sigma A$ in any Kripke model, then also $w \Vdash^\sigma \Gamma \to w \Vdash^\sigma A$ in the model $\mathcal{U}$ above. Since from the $\uparrow$-part of Theorem 2.4 we have that $A, \Gamma \Vdash^\sigma A$, and hence $\Gamma \Vdash^\sigma \Gamma$, then from the $\downarrow$-part of the same theorem there exists a term p such that $\Gamma \vdash p : A$. □

If one wants to extend this technique for proving completeness for Kripke models to the rest of the intuitionistic connectives, $\perp$, $\vee$ and $\exists$, the following meta-mathematical problems appear, which have been investigated in the middle of the last century. At that time, Kreisel, based on observations of Gödel, showed [24][21, Section 3.1.1] that for a wide range of intuitionistic semantics, into which Kripke's can also be fit:

[3]The term "strong completeness by substitution" is used to make a link with [24].

- If one can prove the completeness for the negative fragment of formulae (built using $\wedge, \bot, \Rightarrow, \forall$, and negated atomic formulae, $X \Rightarrow \bot$), of intuitionistic logic, then one can prove Markov's Principle,

$$\neg\neg\exists n A_0(n) \to \exists n A_0(n), \tag{MP}$$

  for $A_0(n)$ decidable. In view of Theorem 2.4, whose proof is purely intuitionistic, this implies that having a completeness proof cover $\bot$ means being able to prove MP, which is known to be independent of many constructive logical systems.

- If one can prove the completeness for all of intuitionistic predicate logic, that is, including $\vee$ and $\exists$, then one can prove a stronger principle,

$$\forall\alpha\neg\neg\exists n A_0(n) \to \forall\alpha\exists n A_0(n), \tag{MP$^+$}$$

  where $A_0(n)$ is decidable and α ranges over functions $\mathbb{N} \to \{0,1\}$.

We mentioned that Veldman [32] used Brouwer's Fan Theorem to handle $\vee$ and $\exists$, but to handle $\bot$ he simply removed the forcing definition "$w \Vdash \bot :=$ false" from the original definition of forcing of Kripke, allowing $w \Vdash \bot$ to be defined, like any other atomic formula, through a (unary) meta-level predicate, $w \Vdash_{\bot}$. In the author's view, Veldman's modification only makes Kripke's original definition more regular: if in Definition 2.1 one considers $\bot$ as a nullary predicate, rather than a logical connective, one obtains Veldman's definition. We remark in passing that it is the same kind of "trick" that allowed Krivine to obtain a constructive version of Gödel's completeness proof for classical predicate logic. [26]

If one tries to straightforwardly extend the NBE-Completeness proof of Theorem 2.4 to cover disjunction (the existential quantifier is analogous), one sees that a problem appears in the case of reflection of sum, $\uparrow^{\tau\vee\sigma}$. There, given a neutral λ-term that derives $\tau\vee\sigma$, one is supposed to prove that $w \Vdash \tau\vee\sigma$ holds, which by definition means to prove that either $w \Vdash \tau$ or $w \Vdash \sigma$ holds. But, since the input λ-term is neutral, it is not of the form $\iota_1 r$ or $\iota_2 r$ (see extended definition of Λ-nf and Λ-ne below), therefore we are not able to decide immediately which of the τ and σ will be proved. Because, a neutral term represents a computation state that is "stuck" due to open variables preventing further reduction.

However, if we consider this situation in a wider context, that of Corollary 2.5, because we are starting with *closed* λ-terms, we know that all of the free variables that appear in the neutral term in question, will be substituted for in the future. Therefore, if we could postpone the decision "$w \Vdash \tau$ or $w \Vdash \sigma$" in the case of $\uparrow^{\tau\vee\sigma}$, we could decide later whether the left or the right disjunct has been proved.

This is precisely what was realised in Computer Science by Danvy, who gave a so-called type-directed partial evaluation algorithm (TDPE) for $\lambda^{\to\vee}$. A TDPE algorithm is another name for an NBE algorithm, although the two concepts were discovered in separate fields of research. [14] Danvy used for this purpose the delimited control operators "shift and reset", that he previously developed with Filinski. [12] Computationally, shift and reset are powerful, because they permit to express any so-called monadic computational side-effect. [15]

For a formal definition of their computational behaviour, we refer the reader to the original articles [12, 13]. Here we will attempt a brief informal description of their semantics.

These delimited control operators consist of two components, the delimiter $\#p$ ("reset") and the operator $\mathcal{S}k.p$ ("shift"). A shift must appear inside the scope of at least one reset, that is, we consider expressions of the form $E[\#F[\mathcal{S}k.p]]$, where $F[\cdot]$ and $E[\cdot]$ are λ-terms with a hole, called evaluation contexts. The reset delimits an evaluation context $F[\cdot]$, allowing shift to gain control of it through the variable k. Shift is a kind of λ-abstraction that allows to manipulate its environment up to the nearest delimiter set by reset.

We give two examples of reduction sequences in a lambda calculus extended with numbers and addition (written in infix-notations). The first one,

$$1 + \#2 + \mathcal{S}k.4 \quad \to \quad 1 + \#4\,\{\,(\lambda a.\#2 + a)/\,k\} \quad = \quad 1 + \#4 \quad \to \quad 1 + 4 \to 5,$$

is an example of using shift which does not make use of its k. This has the effect of simply replacing the evaluation context with the current term under the shift. In the second example,

$$\begin{aligned} & 1 + \#2 + \mathcal{S}k.k4 + k8 \\ \to & 1 + \#(\lambda a.\#2 + a)4 + (\lambda a.\#2 + a)8 \\ \to^{+} & 1 + \#(\#6) + (\#10) \\ \to^{+} & 1 + \#6 + 10 \\ \to^{+} & 17, \end{aligned}$$

shift uses its k twice, having as effect the double use of the evaluation context "$2 + \cdot$ ".

A similar usage of shift happens in Danvy's NBE algorithm for $\lambda^{\to\vee}$,

$$\begin{aligned} \downarrow^{\tau} &: D \to \Lambda\text{-nf} & \\ \downarrow^{\tau} &:= a \mapsto a & \tau\text{-atomic} \\ \downarrow^{\tau\to\sigma} &:= S \mapsto \lambda a.\#\downarrow^{\sigma} (S\cdot \uparrow^{\tau} a) & a\text{-fresh} \\ \downarrow^{\tau\vee\sigma} &:= S \mapsto \begin{cases} \iota_1(\downarrow^{\tau} S') & ,\ \text{if } S = \mathrm{inl}\cdot S' \\ \iota_2(\downarrow^{\sigma} S') & ,\ \text{if } S = \mathrm{inr}\cdot S' \end{cases} & \\ \uparrow^{\tau} &: \Lambda\text{-ne} \to D & \\ \uparrow^{\tau} &:= a \mapsto a & \tau\text{-atomic} \\ \uparrow^{\tau\to\sigma} &:= e \mapsto S \mapsto \uparrow^{\sigma} e(\downarrow^{\tau} S) & \\ \uparrow^{\tau\vee\sigma} &:= e \mapsto \mathcal{S}\kappa.\mathsf{case}\ e\ \mathsf{of}\ \begin{array}{c} (a_1.\#\kappa\cdot(\mathrm{inl}\cdot(\uparrow^{\tau} a_1))\ \| \\ a_2.\#\kappa\cdot(\mathrm{inr}\cdot(\uparrow^{\sigma} a_2))) \end{array} & a_i\text{-fresh} \end{aligned}$$

where we extend the inductive definition of normal and neutral λ-terms as follows:

$$\Lambda\text{-nf} \ni r := e^\tau \mid \lambda a^\tau.r^\sigma \mid \iota_1^\tau r \mid \iota_2^\tau r$$
$$\Lambda\text{-ne} \ni e := a^\tau \mid e^{\tau\to\sigma} r^\tau \mid \mathsf{case}\ e^{\tau\vee\sigma}\ \mathsf{of}\ (a_1^\tau.r_1^\rho \| a_2^\sigma.r_2^\rho)$$

For the computational behaviour of the algorithm, we point the reader to the original article [10].

One may wonder if using these computational facilities is safe from the logical point of view, that is, whether Danvy's algorithm constitutes at the same time a constructive proof of completeness of intuitionistic predicate logic with disjunction (and, by analogy, with $\exists$) with respect to Kripke models. In order to affirm that, we need a typing system for shift and reset that at the same time is a constructive logic and is able to type-check the above computer program. While we do know how to construct typing systems for shift and reset that are constructive, [18, 22] more work is necessary in order to make one that can type-check the program. On the other hand, the existing typing systems are either so-called type-and-effect system [12, 2], where implication is a quaternary not binary connective, or variants of classical logic [29].

3. Kripke-CPS Models and Their Completeness

Historically, shift and reset appeared as a way to write in direct style all the programs that required to be written in continuation-passing style (CPS). We can thus hope to give a normalisation-by-evaluation proof for intuitionistic logic with $\vee$ and $\exists$ in continuation-passing style, or, logically speaking, because CPS translations are analogous to double-negation translations, to give an "indirect" such proof of Completeness. Note, however, that this indirectness does not make the proof less constructive: we do get a constructive Completeness proof, and one can use it to compute normal forms of terms,[4] albeit one can not keep the same notion of model as Kripke models.

In this section we present a notion of model that we developed following this idea, by suitably inserting continuations into the notion of Kripke model. We prove that the new models are sound and complete for full intuitionistic predicate logic.

Definition 3.1. An *Intuitionistic Kripke-CPS model (IK-CPS)* is given by:

- a preorder $(K, \leq)$ of *possible worlds*;
- a $\boxed{\text{binary}}$ relation $(-) \Vdash_\perp^{(-)}$ between worlds and formulae, labelling a world as *exploding*[5] *at a formula*;

[4] Even in practice, using the formalisation in the Coq prof assistant. [20]

[5] The term "exploding" nodes is somewhat of a folklore, and is used besides the terms "fallible" nodes and "sick" nodes. The name comes from the original purpose of this predicate, to stand for forcing of $\perp$.

- a binary relation $(-) \Vdash_s (-)$ of *strong forcing* between worlds and *atomic* formulae (predicates), such that

$$\text{for all } w' \geq w, w \Vdash_s X \subseteq w' \Vdash_s X,$$

- and a domain of quantification $D(w)$ for each world w, such that

$$\text{for all } w' \geq w, D(w) \subseteq D(w').$$

The relation $(-) \Vdash_s (-)$ of *strong forcing* is *extended from atomic to composite formulae* inductively and by simultaneously defining $\boxed{\text{one}}$ new relation, (non-strong) forcing:

⋆ A formula A is *forced* at world w (notation $w \Vdash^\sigma A$) if, $\boxed{\text{for any formula } C,}$

$$\forall w' \geq w.\ \left(\forall w'' \geq w'.\ w'' \Vdash_s^\sigma A \to w'' \Vdash_\perp^C\right) \to w' \Vdash_\perp^C;$$

- $w \Vdash_s^{\{x_1 \mapsto d_1, \ldots, x_n \mapsto d_n\}} X(x_1, \ldots, x_n)$ if $(w \Vdash_s X)(d_1, \ldots, d_n)$;
- $w \Vdash_s^\sigma A \wedge B$ if $w \Vdash^\sigma A$ and $w \Vdash^\sigma B$;
- $w \Vdash_s^\sigma A \vee B$ if $w \Vdash^\sigma A$ or $w \Vdash^\sigma B$;
- $w \Vdash_s^\sigma A \Rightarrow B$ if for all $w' \geq w$, $w' \Vdash^\sigma A$ implies $w' \Vdash^\sigma B$;
- $w \Vdash_s^\sigma \forall x.A(x)$ if for all $w' \geq w$ and all $d \in D(w')$, $w' \Vdash^{\sigma, x \mapsto d} A(x)$;
- $w \Vdash_s^\sigma \exists x.A(x)$ if $w \Vdash^{\sigma, x \mapsto d} A(x)$ for some $d \in D(w)$.

Remark 3.2. Certain details of the definition have been put into boxes to facilitate the comparison carried out in Section 5.

Remark 3.3. We use explicit environments σ for handling quantifiers, for the reasons mentioned in Remark 2.2. To lessen the notational burden, we will skip writing the σ-superscript when it is not relevant, like in the non-quantifier cases of the proof of Theorem 3.13.

Remark 3.4. In the definition of (non-strong) forcing, there is a universal quantification over all formulae C. We could as well have used a quantification over $\mathbb{N}$ and work with encodings of formulae when constructing the universal model of Definition 3.10. What is important, however, is to remark that this quantification if *first-order*, the C-s are individuals not predicates or types.

Remark 3.5. The condition "for all formula C" is only necessary for the soundness proof (Theorem 3.9) to go through, more precisely, the cases of elimination rules for $\vee$ and $\Rightarrow$. The completeness proof (Theorem 3.13) goes through even if we define $w \Vdash A$ by

$$\forall w' \geq w.\ \left(\forall w'' \geq w'.\ w'' \Vdash_s A \to w'' \Vdash_\perp^A\right) \to w' \Vdash_\perp^A .$$

Lemma 3.6. *Strong forcing and (non-strong) forcing are monotone in any IK-CPS model, that is, given $\sigma \in D(w)$ and $w' \geq w$, $w \Vdash_s^\sigma A$ implies $w' \Vdash_s^\sigma A$, and $w \Vdash^\sigma A$ implies $w' \Vdash^\sigma A$.*

Proof. Monotonicity of strong forcing is proved by induction on the complexity of the formula, while that of forcing is by definition. The proof is easy and available in the Coq formalisation.

It is not needed that the exploding node predicate be monotone. □

Lemma 3.7. *The following "monadic" operations are definable for IK-CPS models:*

"unit" $\eta(\cdot)$ $\quad w \Vdash_s^\sigma A \to w \Vdash^\sigma A$

"bind" $(\cdot)^*(\cdot)$ $\quad (\forall w' \geq w.\ w' \Vdash_s^\sigma A \to w' \Vdash^\sigma B) \to w \Vdash^\sigma A \to w \Vdash^\sigma B$

Proof. Easy, using Lemma 3.6. If we leave implicit the handling of formulae C, worlds, and monotonicity, we have the following procedures behind the proofs.

$$\eta(\alpha) = \kappa \mapsto \kappa \cdot \alpha$$
$$(\phi)^*(\alpha) = \kappa \mapsto \alpha \cdot (\beta \mapsto \phi \cdot \beta \cdot \kappa)$$

□

Remark 3.8. These operations are called monadic because their typing satisfies the functional programming notion of monad. We currently do not know if a corresponding notion of monad in Category Theory arises from them.

With Table 1, we fix a derivation system and proof term notation for minimal intuitionistic predicate logic. There are two kinds of variables, proof term variables $a, b, \ldots$, and individual (quantifier) variables $x, y, \ldots$. Individual terms (constants) are denoted by t. We rely on these conventions to resolve the apparent ambiguity of the syntax: the abstraction $\lambda a.p$ is a proof term for $\Rightarrow_I$, while $\lambda x.p$ is a proof term for $\forall_I$; (p, q) is a proof term for $\wedge_I$, while (t, q) is a proof term for $\exists_I$.

We supplement the characterisation of normal and neutral terms from page 7:

$$\begin{aligned}
\Lambda\text{-nf} \ni r :=& e \mid \lambda a.r \mid \iota_1 r \mid \iota_2 r \mid (r_1, r_2) \mid \lambda x.r \mid (t, r) \\
\Lambda\text{-ne} \ni e :=& a \mid er \mid \mathsf{case}\ e\ \mathsf{of}\ (a_1.r_1 \| a_2.r_2) \mid \pi_1 e \mid \pi_2 e \mid et \mid \\
& \mathsf{dest}\ e\ \mathsf{as}\ (x.a)\ \mathsf{in}\ r
\end{aligned}$$

As before, let $w \Vdash^\sigma \Gamma$ denote that all formulae from Γ are forced.

Theorem 3.9 (Soundness)**.** *If $\Gamma \vdash p : A$, then, in any world w of any IK-CPS model, and for any $\sigma \in D(w)$, if $w \Vdash^\sigma \Gamma$, then $w \Vdash^\sigma A$.*

Proof. The proof is by induction on the height of the derivation tree. We consider the last used derivation rule. Here we give the propositional case for $\Rightarrow_E$, and the quantifier cases. For the rest of the cases from the propositional fragment, one can be guided by the algorithm given in Section 4.

$$\frac{(a : A) \in \Gamma}{\Gamma \vdash a : A}\,\mathrm{Ax}$$

$$\frac{\Gamma \vdash p : A_1 \qquad \Gamma \vdash q : A_2}{\Gamma \vdash (p, q) : A_1 \wedge A_2}\,\wedge_I \qquad\qquad \frac{\Gamma \vdash p : A_1 \wedge A_2}{\Gamma \vdash \pi_i p : A_i}\,\wedge_E^i$$

$$\frac{\Gamma \vdash p : A_i}{\Gamma \vdash \iota_i p : A_1 \vee A_2}\,\vee_I^i$$

$$\frac{\Gamma \vdash p : A_1 \vee A_2 \qquad \Gamma, a_1 : A_1 \vdash q_1 : C \qquad \Gamma, a_2 : A_2 \vdash q_2 : C}{\Gamma \vdash \mathsf{case}\ p\ \mathsf{of}\ \ (a_1.q_1 \| a_2.q_2) : C}\,\vee_E$$

$$\frac{\Gamma, a : A_1 \vdash p : A_2}{\Gamma \vdash \lambda a.p : A_1 \Rightarrow A_2}\,\Rightarrow_I \qquad\qquad \frac{\Gamma \vdash p : A_1 \Rightarrow A_2 \qquad \Gamma \vdash q : A_1}{\Gamma \vdash pq : A_2}\,\Rightarrow_E$$

$$\frac{\Gamma \vdash p : A(x) \qquad x\text{-fresh}}{\Gamma \vdash \lambda x.p : \forall x.A(x)}\,\forall_I \qquad\qquad \frac{\Gamma \vdash p : \forall x.A(x)}{\Gamma \vdash pt : A(t)}\,\forall_E$$

$$\frac{\Gamma \vdash p : A(t)}{\Gamma \vdash (t, p) : \exists x.A(x)}\,\exists_I$$

$$\frac{\Gamma \vdash p : \exists x.A(x) \qquad \Gamma, a : A(x) \vdash q : C \qquad x\text{-fresh}}{\Gamma \vdash \mathsf{dest}\ p\ \mathsf{as}\ \ (x.a)\ \ \mathsf{in}\ q : C}\,\exists_E$$

Table 1: Proof term annotation for the natural deduction system of minimal intuitionistic predicate logic (MQC)

Case $\Rightarrow_E$. To show $w \Vdash \Gamma \to w \Vdash B$, suppose $w \Vdash \Gamma$, and suppose $C, w' \geq w$ are given, and let κ' denote a proof of

$$\forall w'' \geq w'.\ (w'' \Vdash_s B \to w'' \Vdash_\perp^C).$$

To show that $w' \Vdash_\perp^C$, apply the induction hypothesis $w \Vdash A \Rightarrow B$ setting in it $C := C, w' := w'$. Now, given ϕ, which denotes a proof of

$$\forall w'' \geq w'.\ (w'' \Vdash A \to w'' \Vdash B),$$

we have to show $w' \Vdash_\perp^C$. We can finish the proof by using ϕ, the other induction hypothesis $w \Vdash A$, and κ'. $\square$

Case $\forall_I$. To show that $w \Vdash^{\sigma} \forall x.A(x)$, we use "unit" of Lemma 3.7, and then we have to show that, for a given $w' \geq w$, $d \in D(w')$, we have that $w' \Vdash^{\sigma,x\mapsto d} A(x)$. But, this is immediate by the induction hypothesis.

Case $\forall_E$. We show that $w \Vdash^{\sigma,x\mapsto d} A(x)$ in the same way as the case of $\Rightarrow_E$, except that ϕ stands for a proof of

$$\forall w'' \geq w'.\ \forall d \in D(w'').\ w'' \Vdash^{\sigma,x\mapsto d} A(x),$$

and there is no (need of a) second induction hypothesis to apply.

Case $\exists_I$. To show that $w \Vdash^{\sigma} \exists x.A(x)$ from the induction hypothesis $w \Vdash^{\sigma,x\mapsto d} A(x)$, where $d \in D(w)$, is immediate by "unit" of Lemma 3.7.

Case $\exists_E$. To show that $w \Vdash C$, suppose that a formula C' and a world $w' \geq w$ are given, and denote by κ' a proof of

$$\forall w'' \geq w'.\ (w'' \Vdash^{\sigma}_{s} C \to w'' \Vdash^{C'}_{\perp}).$$

To show that $w' \Vdash^{C'}_{\perp}$, apply the induction hypothesis $w' \Vdash \exists x.A(x)$ setting $C := C', w' := w'$, and then, given $d \in D(w')$ and a proof of $w' \Vdash^{\sigma,x\mapsto d} A(x)$, show $w' \Vdash^{C'}_{\perp}$ by using the other induction hypothesis and κ'.

Definition 3.10. The *Universal IK-CPS model $\mathcal{U}$* is obtained by setting:

- K to be the set of contexts Γ of MQC;
- $\Gamma \leq \Gamma'$ iff $\Gamma \subseteq \Gamma'$;
- $\Gamma \Vdash_{s} X$ iff there is a derivation in normal form of $\Gamma \vdash X$ in MQC, where X is an atomic formula;
- $\Gamma \Vdash^{C}_{\perp}$ iff there is a derivation in normal form of $\Gamma \vdash C$ in MQC;
- for any w, $D(w)$ is a set of individuals for MQC (that is, $D(-)$ is a constant function from worlds to sets of individuals).

$(-) \Vdash_{s} (-)$ is monotone for the first argument because of the weakening property for intuitionistic "$\vdash$".

Remark 3.11. The difference between strong forcing "$\Vdash_{s}$" and the exploding node predicate "$\Vdash^{C}_{\perp}$" in $\mathcal{U}$ is that the former is defined on atomic formulae, while the latter is defined on any kind of formulae. Although, for $\mathcal{U}$, it would suffice to use "$\Vdash^{C}_{\perp}$" for "$\Vdash_{s}$", we do not want to make the abstract definition of IK-CPS model less general by unifying the two. For similar reasons, we keep $D(-)$ a monotone function on worlds instead of the constant one used for $\mathcal{U}$.

Lemma 3.12. *We can also define the monadic* "run" *operation on the universal model $\mathcal{U}$, but only for* atomic *formulae X:*

$$\mu(\cdot) : w \Vdash X \to w \Vdash_{s} X.$$

Proof. By setting $C := X$ and applying reflexivity of the preorder and the identity function. $\square$

Theorem 3.13 (Completeness for $\mathcal{U}$)**.** *For any closed formula A and closed context Γ, the following hold for $\mathcal{U}$:*

$$\Gamma \Vdash A \longrightarrow \text{ there exists } r \in \Lambda\text{-}nf \text{ such that } \Gamma \vdash r : A \qquad (\downarrow)$$

$$\text{for } e \in \Lambda\text{-}ne, \Gamma \vdash e : A \longrightarrow \Gamma \Vdash A \qquad (\uparrow)$$

Proof. We prove simultaneously the two statements by induction on the complexity of formula A.

We skip writing the proof term annotations, and write just $\Gamma \vdash A$ instead of "there exists p such that $\Gamma \vdash p : A$", in order to decrease the level of detail. The algorithm behind this proof that concentrates on proof terms is given in Section 4.

Base case. $(\downarrow)$ is by "run" (Lemma 3.12), $(\uparrow)$ is by "unit" (Lemma 3.7).

Induction case for $\wedge$. Let $\Gamma \Vdash A \wedge B$ i.e.

$$\forall C.\ \forall \Gamma' \geq \Gamma.\ \left((\forall \Gamma'' \geq \Gamma'.\ \Gamma'' \Vdash A \text{ and } \Gamma'' \Vdash B \to \Gamma'' \vdash C) \to \Gamma' \vdash C \right).$$

We apply this hypothesis by setting $C := A \wedge B$ and $\Gamma' := \Gamma$, and then, given $\Gamma'' \geq \Gamma$ s.t. $\Gamma'' \Vdash A$ and $\Gamma'' \Vdash B$, we have to derive $\Gamma'' \vdash A \wedge B$. But, this is immediate by applying the $\wedge_I$ rule and the induction hypothesis $(\downarrow)$ twice, for A and for B.

Let $\Gamma \vdash A \wedge B$ be a neutral derivation. We prove $\Gamma \Vdash A \wedge B$ by applying unit (Lemma 3.7), and then applying the induction hypothesis $(\downarrow)$ on $\wedge_I^1$, $\wedge_I^2$, and the hypothesis.

Induction case for $\vee$. Let $\Gamma \Vdash A \vee B$ i.e.

$$\forall C.\ \forall \Gamma' \geq \Gamma.\ \left((\forall \Gamma'' \geq \Gamma'.\ \Gamma'' \Vdash A \text{ or } \Gamma'' \Vdash B \to \Gamma'' \vdash C) \to \Gamma' \vdash C \right).$$

We apply this hypothesis by setting $C := A \vee B$ and $\Gamma' := \Gamma$, and then, given $\Gamma'' \geq \Gamma$ s.t. $\Gamma'' \Vdash A$ or $\Gamma'' \Vdash B$, we have to derive $\Gamma'' \vdash A \vee B$. But, this is immediate, after a case distinction, by applying the $\vee_I^i$ rule and the induction hypothesis $(\downarrow)$.

We now consider the only case (besides $\uparrow^{\exists x A(x)}$ below) where using shift and reset, or our Kripke-style models, is crucial. Let $\Gamma \vdash A \vee B$ be a neutral derivation. Let a formula C and $\Gamma' \geq \Gamma$ be given, and let

$$\forall \Gamma'' \geq \Gamma'.\ (\Gamma'' \Vdash A \text{ or } \Gamma'' \Vdash B \to \Gamma'' \vdash C). \qquad (\#)$$

We prove $\Gamma' \vdash C$ by the following derivation tree:

$$\dfrac{\dfrac{\Gamma \vdash A \vee B}{\Gamma' \vdash A \vee B} \qquad \dfrac{\dfrac{\dfrac{\dfrac{A \in A, \Gamma'}{A, \Gamma' \vdash A}\,\text{Ax}}{A, \Gamma' \Vdash A}\,(\uparrow)}{A, \Gamma' \Vdash A \text{ or } A, \Gamma' \Vdash B}\,\text{inl}}{A, \Gamma' \vdash C}\,(\#) \qquad \dfrac{\dfrac{\dfrac{\dfrac{B \in B, \Gamma'}{B, \Gamma' \vdash B}\,\text{Ax}}{B, \Gamma' \Vdash B}\,(\uparrow)}{B, \Gamma' \Vdash A \text{ or } B, \Gamma' \Vdash B}\,\text{inr}}{B, \Gamma' \vdash C}\,(\#)}{\Gamma' \vdash C}\,\vee_E$$

Induction case for $\Rightarrow$. Let $\Gamma \Vdash A \Rightarrow B$ i.e.

$$\forall C.\ \forall \Gamma' \geq \Gamma.$$
$$((\forall \Gamma'' \geq \Gamma'.\ (\forall \Gamma_3 \geq \Gamma''.\ \Gamma_3 \Vdash A \to \Gamma_3 \Vdash B) \to \Gamma'' \vdash C) \to \Gamma' \vdash C)\,.$$

We apply this hypothesis by setting $C := A \Rightarrow B$ and $\Gamma' := \Gamma$, and then, given $\Gamma'' \geq \Gamma$ s.t.

$$\forall \Gamma_3 \geq \Gamma''.\ \Gamma_3 \Vdash A \to \Gamma_3 \Vdash B \tag{\#}$$

we have to derive $\Gamma'' \vdash A \Rightarrow B$. This follows by applying $(\Rightarrow_I)$, the IH for($\downarrow$), then $(\#)$, and finally the IH for $(\uparrow)$ with the Ax rule.

Let $\Gamma \vdash A \Rightarrow B$ be a neutral derivation. We prove $\Gamma \Vdash A \Rightarrow B$ by applying unit (Lemma 3.7), and then, given $\Gamma' \geq \Gamma$ and $\Gamma' \Vdash A$, we have to show that $\Gamma' \Vdash B$. This is done by applying the IH for $(\uparrow)$ on the $(\Rightarrow_E)$ rule, with the IH for $(\downarrow)$ applied to $\Gamma' \Vdash A$.

Induction case for $\forall$. We recall that the domain function $D(-)$ is constant in the universal model $\mathcal{U}$.

Let $\Gamma \Vdash^{\sigma} \forall x A(x)$ i.e.

$$\forall C.\ \forall \Gamma' \geq \Gamma.$$
$$\Big(\big(\forall \Gamma'' \geq \Gamma'.\ \big(\forall \Gamma_3 \geq \Gamma''.\ \forall t \in D.\ \Gamma_3 \Vdash^{\sigma, x \mapsto t} A(x)\big) \to \Gamma'' \vdash C\big) \to \Gamma' \vdash C\Big)\,.$$

We apply this hypothesis by setting $C := \forall x A(x)$ and $\Gamma' := \Gamma$, and then, given $\Gamma'' \geq \Gamma$ s.t.

$$\forall \Gamma_3 \geq \Gamma''.\ \forall t \in D.\ \Gamma_3 \Vdash^{\sigma, x \mapsto t} A(x) \tag{\#}$$

we have to derive $\Gamma'' \vdash \forall x A(x)$. This follows by applying $(\forall_I)$, the IH for($\downarrow$), and then $(\#)$.

Let $\Gamma \vdash \forall x A(x)$ be a neutral derivation. We prove $\Gamma \Vdash^{\sigma} \forall x A(x)$ by applying unit (Lemma 3.7), and then, given $\Gamma' \geq \Gamma$ and $t \in D$, we have to show that $\Gamma' \Vdash^{\sigma, x \mapsto t} A(t)$. This is done by applying the IH for $(\uparrow)$ on the $(\forall_E)$ rule and the hypothesis $\Gamma \vdash \forall x A(x)$.

Induction case for $\exists$. Let $\Gamma \Vdash^{\sigma} \exists x A(x)$ i.e.

$$\forall C.\ \forall \Gamma' \geq \Gamma.\ \Big(\big(\forall \Gamma'' \geq \Gamma'.\ \big(\exists t \in D.\ \Gamma'' \Vdash^{\sigma, x \mapsto t} A(t)\big) \to \Gamma'' \vdash C\big) \to \Gamma' \vdash C\Big)\,.$$

We apply this hypothesis by setting $C := \exists x A(x)$ and $\Gamma' := \Gamma$, and then, given $\Gamma'' \geq \Gamma$ s.t. $\exists t \in D.\ \Gamma'' \Vdash^{\sigma, x \mapsto t} A(x)$, we have to derive $\Gamma'' \vdash \exists x A(x)$. This follows by applying $(\exists_I)$ with $t \in D$, and the IH for($\downarrow$).

Let $\Gamma \vdash \exists x A(x)$ be a neutral derivation. Let a formula C and $\Gamma' \geq \Gamma$ be given, and let

$$\forall \Gamma'' \geq \Gamma'. \ \left(\exists t \in D. \Gamma'' \Vdash^{\sigma, x \mapsto t} A(x) \to \Gamma'' \vdash C\right). \tag{\#}$$

We prove $\Gamma' \vdash C$ by the following derivation tree:

$$\dfrac{\dfrac{\Gamma \vdash \exists x A(x)}{\Gamma' \vdash \exists x A(x)} \qquad \dfrac{\dfrac{\dfrac{A(x) \in A(x), \Gamma'}{A(x), \Gamma' \vdash A(x)}\,\mathrm{Ax}}{A(x), \Gamma' \Vdash^{\sigma, x \mapsto t} A(x)}\,(\uparrow)}{A(x), \Gamma' \vdash C}\,(\#) \qquad x\text{-fresh}}{\Gamma' \vdash C}\,\exists_E$$

For all of the ($\downarrow$)*-directions, it holds that* $r \in \Lambda$*-nf.* By verifying that all above proof cases generate derivations in normal form. □

In the same way as for corollaries 2.5 and 2.6, we obtain the following two.

Corollary 3.14. *For every closed proof term* p *of MQC, such that* $\vdash p : A$*, there exists a proof term* p' *in normal form, such that* $\vdash p' : A$*.*

Corollary 3.15. *If in any IK-CPS model, at any world* w*,* $w \Vdash \Gamma$ *implies* $w \Vdash A$*, then there exists a term* p *and a derivation* $\Gamma \vdash p : A$*.*

4. Normalisation by Evaluation in IK-CPS Models

In this section we give the algorithm that we manually extracted from the formalisation in the proof assistant Coq, for the restriction to the interesting propositional fragment that involves implication and disjunction. The algorithm extracted automatically by Coq contains too many details to be instructive, however, the interested reader can directly run it inside the proof assistant.

The following evaluation function for $\lambda^{\to\vee}$-terms is behind the proof of Theorem 3.9:

$$[\![\Gamma \vdash p : A]\!]_{w \Vdash \Gamma} : w \Vdash A$$

$$\begin{aligned}
[\![a]\!]_\rho &:= \rho(a) \\
[\![\lambda a.p]\!]_\rho &:= \kappa \mapsto \kappa \cdot (\alpha \mapsto [\![p]\!]_{\rho, a \mapsto \alpha}) = \eta \cdot (\alpha \mapsto [\![p]\!]_{\rho, a \mapsto \alpha}) \\
[\![pq]\!]_\rho &:= \kappa \mapsto [\![p]\!]_\rho \cdot (\phi \mapsto \phi \cdot [\![q]\!]_\rho \cdot \kappa) \\
[\![\iota_1 p]\!]_\rho &:= \kappa \mapsto \kappa \cdot (\mathrm{inl} \cdot [\![p]\!]_\rho) = \eta \cdot (\mathrm{inl} \cdot [\![p]\!]_\rho) \\
[\![\iota_2 p]\!]_\rho &:= \kappa \mapsto \kappa \cdot (\mathrm{inr} \cdot [\![p]\!]_\rho) = \eta \cdot (\mathrm{inr} \cdot [\![p]\!]_\rho) \\
[\![\mathsf{case}\ p\ \mathsf{of}\ (a_1.q_1 \| a_2.q_2)]\!]_\rho &:= \kappa \mapsto [\![p]\!]_\rho \cdot \left(\gamma \mapsto \begin{cases} [\![q_1]\!]_{\rho, a_1 \mapsto \alpha} \cdot \kappa & , \gamma = \mathrm{inl} \cdot \alpha \\ [\![q_2]\!]_{\rho, a_2 \mapsto \beta} \cdot \kappa & , \gamma = \mathrm{inr} \cdot \beta \end{cases}\right)
\end{aligned}$$

The following is the algorithm behind Theorem 3.13:

$$\downarrow^A_\Gamma : \Gamma \Vdash A \to \{p \in \Lambda\text{-nf} \ \mid \Gamma \vdash p : A\}$$
$$\uparrow^A_\Gamma : \{e \in \Lambda\text{-ne} \ \mid \Gamma \vdash e : A\} \to \Gamma \Vdash A$$

$$\begin{aligned}
\downarrow^X_\Gamma &:= \alpha \mapsto \mu \cdot \alpha && X\text{-atomic}\\
\uparrow^X_\Gamma &:= e \mapsto \eta \cdot e && X\text{-atomic}\\
\downarrow^{A\Rightarrow B}_\Gamma &:= \eta \cdot \big(\phi \mapsto \lambda a.\downarrow^B_{\Gamma,a:A} \big(\phi \cdot \uparrow^A_{\Gamma,a:A} a\big)\big) && a\text{-fresh}\\
\uparrow^{A\Rightarrow B}_\Gamma &:= e \mapsto \eta \cdot \big(\alpha \mapsto \uparrow^B_\Gamma \big(e\,(\downarrow^A_\Gamma \alpha)\big)\big) &&\\
\downarrow^{A\vee B}_\Gamma &:= \eta \cdot \left(\gamma \mapsto \left\{ \begin{array}{ll} \iota_1 \downarrow^A_\Gamma \alpha & \text{if } \gamma = \mathrm{inl} \cdot \alpha \\ \iota_2 \downarrow^B_\Gamma \beta & \text{if } \gamma = \mathrm{inr} \cdot \beta \end{array}\right. \right) &&\\
\uparrow^{A\vee B}_\Gamma &:= e \mapsto \kappa \mapsto \mathsf{case}\ e\ \mathsf{of}\ \begin{array}{l} \big(a_1.\kappa \cdot \big(\mathrm{inl} \cdot \uparrow^A_{\Gamma,a_1:A} a_1\big) \parallel \\ \ a_2.\kappa \cdot \big(\mathrm{inr} \cdot \uparrow^B_{\Gamma,a_2:B} a_2\big)\big) \end{array} && a_i\text{-fresh}
\end{aligned}$$

5. Variants and Relation to Classical Kripke Models

5.1. *"Call-by-value" Models*

Defining forcing on composite formulae in Definition 3.1 proceeds analogously to defining the call-by-name CPS translation [30], or Kolmogorov's double-negation translation [31, 28]. A definition analogous to the call-by-value CPS translation [30], i.e. Kuroda's double-negation translation, is also possible, by defining (non-strong) forcing by:

- $w \Vdash_s^{\{x_1 \mapsto d_1, \ldots, x_n \mapsto d_n\}} X(x_1, \ldots, x_n)$ if $(w \Vdash_s X)(d_1, \ldots, d_n)$;
- $w \Vdash_s^{\sigma} A \wedge B$ if $w \Vdash_s^{\sigma} A$ and $w \Vdash_s^{\sigma} B$;
- $w \Vdash_s^{\sigma} A \vee B$ if $w \Vdash_s^{\sigma} A$ or $w \Vdash_s^{\sigma} B$;
- $w \Vdash_s^{\sigma} A \Rightarrow B$ if for all $w' \geq w$, $w \Vdash_s^{\sigma} A$ implies $w \Vdash B$;
- $w \Vdash_s^{\sigma} \forall x.A(x)$ if for all $w' \geq w$ and all $d \in D(w')$, $w' \Vdash^{\sigma, x \mapsto d} A(x)$;
- $w \Vdash_s^{\sigma} \exists x.A(x)$ if $w \Vdash_s^{\sigma, x \mapsto d} A(x)$ for some $d \in D(w)$.

One can prove this variant of IK-CPS models sound and complete, similarly to Section 3, except that, in the statement of Soundness, one needs to put $w \Vdash_s \Gamma$ in place of $w \Vdash \Gamma$.

5.2. *Classical Models*

In [20, 21, 23], based on work with Lee and Herbelin, we presented the following notion of model which is complete for *classical* predicate logic and represents an NBE algorithm for it.

Definition 5.1. A *Classical Kripke-CPS model (CK-CPS)*, is given by:

- a preorder $(K, \leq)$ of *possible worlds*;
- a $\boxed{\text{unary}}$ relation on worlds $(-) \Vdash_{\perp}$ labelling a world as *exploding*;
- a binary relation $(-) \Vdash_{s} (-)$ of *strong forcing* between worlds and atomic formulae, such that

$$\text{for all } w' \geq w, w \Vdash_s X \subseteq w' \Vdash_s X,$$

- and a domain of quantification $D(w)$ for each world w, such that

$$\text{for all } w' \geq w, D(w) \subseteq D(w').$$

The relation $(-) \Vdash_s (-)$ of *strong forcing* is *extended from atomic to composite formulae* inductively and by simultaneously defining $\boxed{\text{two}}$ new relations, refutation and (non-strong) forcing:

- ⋆ A formula A is *refuted* in the world w (notation $w : A \Vdash$) if any world $w' \geq w$, which strongly forces A, is exploding;
- ⋆ A formula A is *forced* in the world w (notation $w \Vdash A$) if any world $w' \geq w$, which refutes A, is exploding;
- $w \Vdash_s^{\{x_1 \mapsto d_1, \ldots, x_n \mapsto d_n\}} X(x_1, \ldots, x_n)$ if $(w \Vdash_s X)(d_1, \ldots, d_n)$;
- $w \Vdash_s^{\sigma} A \wedge B$ if $w \Vdash^{\sigma} A$ and $w \Vdash^{\sigma} B$;
- $w \Vdash_s^{\sigma} A \vee B$ if $w \Vdash^{\sigma} A$ or $w \Vdash^{\sigma} B$;
- $w \Vdash_s^{\sigma} A \Rightarrow B$ if for all $w' \geq w$, $w' \Vdash^{\sigma} A$ implies $w' \Vdash^{\sigma} B$;
- $w \Vdash_s^{\sigma} \forall x. A(x)$ if for all $w' \geq w$ and all $d \in D(w')$, $w' \Vdash^{\sigma, x \mapsto d} A(x)$;
- $w \Vdash_s^{\sigma} \exists x. A(x)$ if $w \Vdash^{\sigma, x \mapsto d} A(x)$ for some $d \in D(w)$.

The differences between Definition 3.1 and Definition 5.1 are marked with boxes. We can also present CK-CPS using binary exploding nodes, by defining $w \Vdash_s \perp := \forall C. w \Vdash_{\perp}^{C}$. Then, we get the following statement of forcing in CK-CPS,

$$\forall w' \geq w. \ \left(\forall w'' \geq w'. \ w'' \Vdash_s^{\sigma} A \to \forall I. w'' \Vdash_{\perp}^{I}\right) \to \forall O. w' \Vdash_{\perp}^{O},$$

versus forcing in IK-CPS,

$$\forall C. \ \forall w' \geq w. \ \left(\forall w'' \geq w'. \ w'' \Vdash_s^{\sigma} A \to w'' \Vdash_{\perp}^{C}\right) \to w' \Vdash_{\perp}^{C} .$$

The difference between forcing in the intuitionistic and classical models is, then, that: 1) the dependency on C is necessary in the intuitionistic case, while it is optional in the classical case; 2) the continuation (the internal implication) in classical forcing is allowed to change the parameter C upon application, whereas in intuitionistic forcing the parameter is not local to the continuation, but to the continuation of the continuation.

The reader may find it instructive to check for himself why Peirce's Law cannot be forced in IK-CPS, while it can be forced in CK-CPS.

6. Conclusion

We gave a constructive (intuitionistic) proof of completeness of MQC with respect to IK-CPS models, inspired by a particular use of delimited control operators. This does not defy Gödel's and Kreisel's "negative" meta-mathematical results of Section 2: simply, their arguments rely on a different class of semantics – intuitionistic semantics *à la* Tarski.[6]

On the other hand, precisely those "negative" results were an indication that one can build constructive systems based on delimited control operators that can prove extra-intuitionistic principles, like the predicate-logic versions of Markov's Principle [18] or Double-negation Shift [22].

Another question that could be asked is whether the results of this paper could simply be obtained by applying the double-negation translation to the classical proof of completeness for Kripke models. While it is quite possible that a constructive completeness proof for full intuitionistic logic (without $\bot$) could be obtained in such a way, like for completeness of classical logic by Krivine [26, 4], it is clear that that would give a different method of proof. Namely, the classical proofs from [25, 31] are Henkin-style proofs which depend on an enumeration of formulae and a constructive version of the ultra-filter theorem that relies on that enumeration [21, Chapter 1], hence, the obtained constructive completeness could not be expected to produce an NBE algorithm, because the later should depend crucially on structural recursion over the type, like our completeness proof, not on an *ad hoc* enumeration.

We pointed out that our algorithm is β-NBE, because were we able to identify $\beta\eta$-equal terms of $\lambda^{\to\vee}$ through our NBE function, we would have solved the problem of the existence of canonical η-long normal form for $\lambda^{\to\vee}$. However, as shown by [17], due to the connection with Tarski's High School Algebra Problem [7, 33], the notion of such a normal form is not finitely axiomatisable. If one looks at examples of $\lambda^{\to\vee}$-terms which are $\beta\eta$-equal but are not normalised to the same term by Danvy's (and our) algorithm, one can see that in the Coq type theory these terms are interpreted as denotations that involve commutative cuts.

In recent unpublished work [11], Danvy also developed a version of his NBE algorithm directly in CPS, without using delimited control operators. In [16], Filinski proves the correctness of an NBE algorithm for Moggi's computational λ-calculus, including sums, by also evaluating the input terms in a domain based on continuations. In [3], Barral gives a program for NBE of λ-calculus with sums by just using the exceptions mechanism of a programming language, which is something *a priori* strictly weaker than using delimited control operators.

In [1], Altenkirch, Dybjer, Hofmann, and Scott, give a topos theoretic proof of NBE for a typed λ-calculus with sums, by constructing a sheaf model. The connection between sheaves and Beth semantics[7] is well known. While the proof

[6] The work of [24] appeared before the invention of Kripke models for intuitionistic logic. [25]

[7] We remark that, for the fragment $\{\Rightarrow, \forall, \wedge\}$, NBE can also be seen as completeness for

is constructive, due to their use of topos theory, for us it is not clear how to extract an algorithm from it.

In [27], Macedonio and Sambin present a notion of model for extensions of Basic logic (a sub-structural logic more primitive than Linear logic), which, for intuitionistic logic, appears to be related to our notion of model. However, they demand that their set of worlds K be saturated, while we do not, and we can hence also work with finite models.

Acknowledgements

To Hugo Herbelin for inspiring discussions and, in particular, for suggesting to try polymorphism, viz. Remark 3.5. To Olivier Danvy for suggesting reference [16].

References

[1] Thorsten Altenkirch, Peter Dybjer, Martin Hofmann, and Philip J. Scott. Normalization by evaluation for typed lambda calculus with coproducts. In *LICS*, pages 303–310, 2001.

[2] Kenichi Asai and Yukiyoshi Kameyama. Polymorphic delimited continuations. In *APLAS*, pages 239–254, 2007.

[3] Freiric Barral. Exceptional NbE for sums. In Olivier Danvy, editor, *Informal proceedings of the 2009 Workshop on Normalization by Evaluation, August 15th 2009, Los Angeles, California*, pages 21–30, 2009.

[4] Stefano Berardi and Silvio Valentini. Krivine's intuitionistic proof of classical completeness (for countable languages). *Ann. Pure Appl. Logic*, 129(1-3):93–106, 2004.

[5] U. Berger and P. Oliva. Modified bar recursion and classical dependent choice. In M. Baaz, S.D. Friedman, and J. Kraijcek, editors, *Logic Colloquium '01, Proceedings of the Annual European Summer Meeting of the Association for Symbolic Logic, held in Vienna, Austria, August 6 - 11, 2001*, volume 20 of *Lecture Notes in Logic*, pages 89–107. Springer, 2005.

[6] Ulrich Berger and Helmut Schwichtenberg. An inverse of the evaluation functional for typed lambda-calculus. In *LICS*, pages 203–211. IEEE Computer Society, 1991.

[7] Stanley Burris and Simon Lee. Tarski's high school identities. *The American Mathematical Monthly*, 100(3):231–236, 1993.

Beth semantics, since forcing in Beth and Kripke models is the same thing on that fragment.

[8] Catarina Coquand. From semantics to rules: A machine assisted analysis. In *CSL '93*, volume 832 of *Lecture Notes in Computer Science*, pages 91–105. Springer, 1993.

[9] Catarina Coquand. A formalised proof of the soundness and completeness of a simply typed lambda-calculus with explicit substitutions. *Higher Order Symbol. Comput.*, 15(1):57–90, 2002.

[10] Olivier Danvy. Type-directed partial evaluation. In *POPL*, pages 242–257, 1996.

[11] Olivier Danvy. A call-by-name normalization function for the simply typed lambda-calculus with sums and products. manuscript, 2008.

[12] Olivier Danvy and Andrzej Filinski. A functional abstraction of typed contexts. Technical report, Computer Science Department, University of Copenhagen, 1989. DIKU Rapport 89/12.

[13] Olivier Danvy and Andrzej Filinski. Abstracting control. In *LISP and Functional Programming*, pages 151–160, 1990.

[14] Peter Dybjer and Andrzej Filinski. Normalization and partial evaluation, 2002.

[15] Andrzej Filinski. Representing monads. In *Proceedings of the 21st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, pages 446–457, 1994.

[16] Andrzej Filinski. Normalization by evaluation for the computational lambda-calculus. In Samson Abramsky, editor, *Typed Lambda Calculi and Applications*, volume 2044 of *Lecture Notes in Computer Science*, pages 151–165. Springer Berlin / Heidelberg, 2001.

[17] Marcelo P. Fiore, Roberto Di Cosmo, and Vincent Balat. Remarks on isomorphisms in typed lambda calculi with empty and sum types. *Ann. Pure Appl. Logic*, 141(1-2):35–50, 2006.

[18] Hugo Herbelin. An intuitionistic logic that proves Markov's principle. In *Proceedings, 25th Annual IEEE Symposium on Logic in Computer Science (LICS '10), Edinburgh, UK, 11-14 July 2010*, page N/A. IEEE Computer Society Press, 2010.

[19] Hugo Herbelin and Gyesik Lee. Forcing-based cut-elimination for Gentzen-style intuitionistic sequent calculus. In Hiroakira Ono, Makoto Kanazawa, and Ruy J. G. B. de Queiroz, editors, *WoLLIC*, volume 5514 of *Lecture Notes in Computer Science*, pages 209–217. Springer, 2009.

[20] Danko Ilik. Formalisation of completeness for Kripke-CPS models, 2009. `https://sites.google.com/site/dankoilik/publications/phd-thesis`.

[21] Danko Ilik. *Constructive Completeness Proofs and Delimited Control*. PhD thesis, École Polytechnique, October 2010.

[22] Danko Ilik. Delimited control operators prove double-negation shift. *Annals of Pure and Applied Logic*, 2011. In press.

[23] Danko Ilik, Gyesik Lee, and Hugo Herbelin. Kripke models for classical logic. *Annals of Pure and Applied Logic*, 161(11):1367 – 1378, 2010. Special Issue: Classical Logic and Computation (2008).

[24] Georg Kreisel. On weak completeness of intuitionistic predicate logic. *J. Symb. Log.*, 27(2):139–158, 1962.

[25] Saul Kripke. Semantical considerations on modal and intuitionistic logic. *Acta Philos. Fennica*, 16:83–94, 1963.

[26] Jean-Louis Krivine. Une preuve formelle et intuitionniste du théorème de complétude de la logique classique. *Bulletin of Symbolic Logic*, 2(4):405–421, 1996.

[27] Damiano Macedonio and Giovanni Sambin. From meta-level to semantics via reflection: a model for basic logic and its extensions. Manuscript.

[28] Chetan Murthy. *Extracting Classical Content from Classical Proofs*. PhD thesis, Department of Computer Science, Cornell University, 1990.

[29] Chetan R. Murthy. Control operators, hierarchies, and pseudo-classical type systems: A-translation at work. In *Proceedings of the ACM SIGPLAN Workshop on Continuations CW92*, pages 49–72. Stanford University, 1992. Technical Report STAN-CS-92-1426.

[30] G. D. Plotkin. Call-by-name, call-by-value and the [lambda]-calculus. *Theoretical Computer Science*, 1(2):125–159, 1975.

[31] A. S. Troelstra and D. van Dalen. *Constructivism in mathematics. Vol. I*, volume 121 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam, 1988. An introduction.

[32] Wim Veldman. An intuitionistic completeness theorem for intuitionistic predicate logic. *J. Symb. Log.*, 41(1):159–166, 1976.

[33] A. J. Wilkie. On exponentiation - a solution to Tarski's high school algebra problem. Technical report, Mathematical Institute, Oxford, UK, 2001.