



HAL
open science

Typed realizability for first-order classical analysis

Valentin Blot

► **To cite this version:**

Valentin Blot. Typed realizability for first-order classical analysis. Logical Methods in Computer Science, 2015, 11 (4), pp.1 - 43. 10.2168/LMCS-11(4:22)2015 . hal-01766871

HAL Id: hal-01766871

<https://hal.science/hal-01766871v1>

Submitted on 14 Apr 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

TYPED REALIZABILITY FOR FIRST-ORDER CLASSICAL ANALYSIS

VALENTIN BLOT

Department of Computer Science, University of Bath, United Kingdom
e-mail address: v.blot@bath.ac.uk

ABSTRACT. We describe a realizability framework for classical first-order logic in which realizers live in (a model of) typed $\lambda\mu$ -calculus. This allows a direct interpretation of classical proofs, avoiding the usual negative translation to intuitionistic logic. We prove that the usual terms of Gödel’s system T realize the axioms of Peano arithmetic, and that under some assumptions on the computational model, the bar recursion operator realizes the axiom of dependent choice. We also perform a proper analysis of relativization, which allows for less technical proofs of adequacy. Extraction of algorithms from proofs of Π_2^0 formulas relies on a novel implementation of Friedman’s trick exploiting the control possibilities of the language. This allows to have extracted programs with simpler types than in the case of negative translation followed by intuitionistic realizability.

INTRODUCTION

Realizability is a mean of formalizing the Brouwer-Heyting-Kolmogorov constructive interpretation of logic. To each formula is associated a set of programs, its realizers, which contain computational information about the formula. While the Curry-Howard isomorphism draws a correspondence between proofs in a certain logical system and programs in a suitable typed programming language, realizability takes a different approach and defines a realizer as a program which behaves like a proof. First, this allows to give computational content to the axioms of a theory, and second this allows to choose more freely the programming language, independently from the logical system. It is even possible to consider untyped programming languages, as was the case in the first realizability model from Kleene [Kle45], in which a realizer may be any recursive function. It is however still possible to consider a typed language, as did Kreisel in his modified realizability model [Kre59]. Both models from Kleene and Kreisel gave computational interpretation to Heyting arithmetic, the intuitionistic variant of Peano arithmetic.

Gödel’s negative translation [Göd33] allowed for the first interpretations of classical logic. Indeed, this translation from classical to intuitionistic logic, when followed by an interpretation of intuitionistic proofs, gives a computational interpretation to classical logic. This is what we call the indirect interpretation. This interpretation can easily be extended

2012 ACM CCS: [Theory of computation]: Logic—Constructive mathematics.

Key words and phrases: Classical realizability, Lambda-mu calculus, Bar-recursion, Axiom of choice.

Research supported by the UK EPSRC grant EP/K037633/1.

to arithmetic, and therefore allows to get computational interpretations of Peano arithmetic from the models of Kleene and Kreisel. Much later, Griffin discovered in [Gri90] that the `call/cc` operator of the functional language Scheme could be typed with a classical principle: the law of Peirce. This opened the possibility to what we call a direct interpretation of classical logic, using programming languages with control features. Following this path, Parigot defined in [Par92] the $\lambda\mu$ -calculus, a language for Gentzen's classical sequent calculus extending the Curry-Howard isomorphism to classical logic. Selinger axiomatized the universal categorical model of $\lambda\mu$ -calculus in [Sel01]. On another side, Krivine considered untyped λ -calculus extended with the `call/cc` operator to give a realizability interpretation to classical second-order Zermelo-Frænkel set theory [Kri01], later extended to handle the axiom of dependent choice [Kri03, Kri09].

In this paper we define a realizability model for first-order classical logic which, contrary to Krivine's and similarly to Kreisel's (but for classical logic), uses typed programs as realizers. We interpret our proofs in the language μ PCF, which is a combination of the functional Turing-complete language PCF with the control features of call-by-name $\lambda\mu$ -calculus. As in Krivine's model, we associate to each formula a truth value and a falsity value which are orthogonal to each other, but we perform a fine analysis of relativization and introduce positive predicates for which only the truth value is required and prove it to be correct through a suitable restriction on the proofs in our logical system. We validate our model by proving that the usual terms of Gödel's system T realize the axioms of Peano arithmetic. We also implement Friedman's trick through the use of an external μ -variable rather than through the replacement of the \perp formula by an existential statement, which allows for a simpler and more effective interpretation. This variable is also used to define the orthogonality relation between our truth and falsity values.

Interpreting the axiom of dependent choice in a classical setting is much more complicated than interpreting arithmetic. Spector defined in [Spe62] the bar recursor and used it in Gödel's Dialectica interpretation [Göd58] (a computational interpretation similar to realizability) to interpret the axiom of countable choice, and therefore the axiom schema of specification. This operator was later studied in [Koh90], and a more uniform version was used in [BBC98] to give an indirect realizability interpretation of countable choice. A version with an implicit termination condition was later defined in [BO05] and used to interpret, still in an indirect realizability setting, the double-negation shift principle and therefore the axiom of dependent choice.

We prove here that under some assumptions on our model of μ PCF, the bar recursion operator of [BO05] realizes the axiom of dependent choice in our direct interpretation of classical logic. Our implementation of Friedman's trick then allows us to obtain an extraction result on Π_2^0 formulas provable in classical analysis (Peano arithmetic + the axiom of dependent choice).

1. LOGIC

We define in this section the logical system under which we will work through the article. First, we define the general case of classical multisorted first-order logic (handling classical reasoning by the use of multi-conclusioned sequents), then we describe the case of logics with equality, the case of Peano arithmetic and its extension with the axiom of dependent choice, and finally we recall some basic definitions about models of classical logic.

1.1. **Classical multisorted first-order logic.** The logical framework we use is multisorted first-order logic, where the sorts are fixed to be the types of simply typed λ -calculus (see e.g. [TVD88]). We build from a set of base sorts ι the set of sorts:

$$\sigma, \tau ::= \iota \mid \sigma \rightarrow \tau$$

which are used for the individuals of the logic. We fix a set of sorted individual constants (ranged over by c^σ) from which we build the set of individuals of the logic:

$$t^\sigma, u^\tau ::= c^\sigma \mid x^\sigma \mid (t^{\sigma \rightarrow \tau} u^\sigma)^\tau$$

We also fix a set of sorted predicates (ranged over by P) from which we define the formulas of the logic:

$$A, B ::= P(t_1^{\sigma_1}, \dots, t_n^{\sigma_n}) \mid \perp \mid A \Rightarrow B \mid A \wedge B \mid \forall x^\sigma A$$

The set of sorts, individuals and formulas of the logic is parameterized by a signature:

Definition 1.1. A signature Σ is a set of base sorts together with a set of sorted constant individuals and a set of sorted predicates.

Negation is defined as $\neg A \triangleq A \Rightarrow \perp$. We choose to have only negative connectives, since the interpretation of our logic in categories of continuations defined in section 2.1.2 is based on a negative call-by-name continuation-passing-style translation. The positive connectives are defined from the negative ones: $A \vee B \triangleq \neg(\neg A \wedge \neg B)$ and $\exists x^\sigma A \triangleq \neg \forall x^\sigma \neg A$. It is well-known that with this coding of positive connectives with negative ones, a formula A is provable in our system if and only if the formula obtained by replacing every $P(t_1, \dots, t_n)$ with $\neg \neg P(t_1, \dots, t_n)$ in A is provable in its intuitionistic restriction. In section 3.1 we will also introduce the notion of negative basic predicates, which are those for which $\neg \neg P \Rightarrow P$ is valid under the realizability interpretation. We perform a more detailed comparison between our system and the usual ones in section 1.3.1.

Since one of the goals of realizability is to provide a computational interpretation of theories beyond pure first-order logic, our model is dependent upon the particular set of axioms under consideration:

Definition 1.2. A theory on a given signature Σ is a set \mathcal{Ax} of closed formulas (axioms) written in the language defined by Σ .

We work in a variant of natural deduction in sequent style, so the interpretation of classical proofs in $\lambda\mu$ -calculus is as direct as possible. In this setting, a context Γ or Δ is a finite unordered sequence of formulas and a sequent is of the form:

$$\Gamma \vdash A \mid \Delta$$

The formula A on the right is the formula that is being worked on, and this presentation is again chosen to have an easy interpretation in $\lambda\mu$ -calculus. The above sequent should be interpreted as: the conjunction of the formulas of Γ implies the disjunction of A and of the formulas of Δ . If Δ is empty we simply write $\Gamma \vdash A$. The set of derivable sequents of a given theory is defined from \mathcal{Ax} using the rules of Figure 1. In this system one can for example derive in any theory the ex-falso principle, the double-negation elimination and Peirce's law:

$$\vdash \perp \Rightarrow A \quad \vdash \neg(\neg A) \Rightarrow A \quad \vdash ((A \Rightarrow B) \Rightarrow A) \Rightarrow A$$

The weakening rule:

$$\frac{\Gamma \vdash A \mid \Delta}{\Gamma' \vdash A \mid \Delta'} \Gamma \subseteq \Gamma', \Delta \subseteq \Delta'$$

$$\begin{array}{c}
\frac{}{\Gamma, A \vdash A \mid \Delta} \quad \frac{}{\Gamma \vdash A \mid \Delta} \quad (A \in \mathcal{A}x) \quad \frac{\Gamma \vdash A \mid A, \Delta}{\Gamma \vdash \perp \mid A, \Delta} \quad \frac{\Gamma \vdash \perp \mid A, \Delta}{\Gamma \vdash A \mid \Delta} \\
\\
\frac{\Gamma, A \vdash B \mid \Delta}{\Gamma \vdash A \Rightarrow B \mid \Delta} \quad \frac{\Gamma \vdash A \Rightarrow B \mid \Delta \quad \Gamma \vdash A \mid \Delta}{\Gamma \vdash B \mid \Delta} \\
\\
\frac{\Gamma \vdash A \mid \Delta \quad \Gamma \vdash B \mid \Delta}{\Gamma \vdash A \wedge B \mid \Delta} \quad \frac{\Gamma \vdash A_1 \wedge A_2 \mid \Delta}{\Gamma \vdash A_i \mid \Delta} \\
\\
\frac{\Gamma \vdash A \mid \Delta}{\Gamma \vdash \forall x^\sigma A \mid \Delta} \quad (x^\sigma \notin \text{FV}(\Gamma, \Delta)) \quad \frac{\Gamma \vdash \forall x^\sigma A \mid \Delta}{\Gamma \vdash A \{t^\sigma/x^\sigma\} \mid \Delta}
\end{array}$$

Figure 1: Deduction rules for first-order multisorted logic

is admissible (α -conversion may be necessary to preserve side condition of \forall -intro) and we use it without mentioning it. The left contraction rule is derivable and the right one is admissible:

$$\frac{\frac{\Gamma, A, A \vdash B \mid \Delta}{\Gamma, A \vdash A \Rightarrow B \mid \Delta} \quad \frac{}{\Gamma, A \vdash A \mid \Delta}}{\Gamma, A \vdash B \mid \Delta} \quad \frac{\frac{\frac{\Gamma \vdash A \mid B, B, \Delta}{\Gamma \vdash A \mid A, B, B, \Delta} \quad B, B, \Delta \subseteq A, B, B, \Delta}{\Gamma \vdash \perp \mid A, B, B, \Delta}}{\Gamma \vdash B \mid A, B, \Delta}}{\Gamma \vdash \perp \mid A, B, \Delta}}{\Gamma \vdash A \mid B, \Delta}$$

so we also use them implicitly. If a sequent $\vdash A$ is derivable in a theory $\mathcal{A}x$, then since $\mathcal{A}x$ may contain infinitely many formulas we write to avoid confusion:

$$\mathcal{A}x \sim A$$

1.2. Equational theories. Most of the theories used in mathematics involve equality, but there are several notions for it. Here we use a primitive Leibniz equality at each type. In fact, as in [Kri09], we use a primitive inequality rather than an equality, which allows us to realize the Leibniz scheme. Moreover, since we are in classical logic, the set of provable sequents is unchanged. In the intuitionistic case however, things are less clear and we discuss it in section 1.3.1.

We say that a theory is an equational theory if it contains for each sort σ an inequality predicate \neq_σ between terms of sort σ (for which we use infix notation), and the following axioms for reflexivity and the Leibniz scheme at all sorts:

$$(\mathit{refl}) \quad \forall x^\sigma (x =_\sigma x) \quad (\mathit{Leib}) \quad \forall \vec{z}^\sigma \forall x^\sigma \forall y^\sigma (\neg A \Rightarrow A \{y/x\} \Rightarrow x \neq_\sigma y)$$

where $t = u \stackrel{\Delta}{=} \neg(t \neq u)$. In this system, the following formulas are derivable:

$$\forall x \forall y (x = y \Rightarrow \forall z (zx = zy)) \quad \forall x \forall y (x = y \Rightarrow \forall z (xz = yz))$$

However, even though the reverse of the implication is provable in PA^ω (see next section 1.3), the reverse of the second one doesn't hold in the general case.

s of sort	$(\sigma \rightarrow \tau \rightarrow \nu) \rightarrow (\sigma \rightarrow \tau) \rightarrow \sigma \rightarrow \nu$	k of sort	$\sigma \rightarrow \tau \rightarrow \sigma$
0 of sort	ι	S of sort	$\iota \rightarrow \iota$
	rec of sort		$\sigma \rightarrow (\iota \rightarrow \sigma \rightarrow \sigma) \rightarrow \iota \rightarrow \sigma$

 Figure 2: Sorted constants of PA^ω

$(refl)$ $\forall x^\sigma (x =_\sigma x)$	$(Leib)$ $\forall \vec{z}^{\vec{\tau}} \forall x^\sigma \forall y^\sigma (\neg A \Rightarrow A \{y/x\} \Rightarrow x \neq_\sigma y)$
(Snz) $\forall x Sx \neq 0$	(ind) $\forall \vec{y} (A \{0/x\} \Rightarrow \forall x (A \Rightarrow A \{Sx/x\}) \Rightarrow \forall x A)$
(Δs) $\forall x \forall y \forall z sx yz = xz (yz)$	(Δk) $\forall x \forall y kxy = x$
$(\Delta rec0)$ $\forall x \forall y recxy 0 = x$	$(\Delta recS)$ $\forall x \forall y \forall z recxy (Sz) = yz (recxy z)$

 Figure 3: Axioms of PA^ω

1.3. Peano arithmetic in finite types: PA^ω . Peano arithmetic in finite types, PA^ω , is an extension of Peano arithmetic in which we can write any term of Gödel's system T and quantify over functions of any type.

The signature Σ_{PA^ω} of Peano arithmetic in finite types contains a single base sort ι and the sorted constants of Figure 2. We will also write 0^σ for the inductively defined term: $0^{\sigma \rightarrow \tau} \triangleq \lambda^{\tau \rightarrow \sigma \rightarrow \tau} 0^\tau$. PA^ω is an equational theory as defined in section 1.2, and inequality is the only predicate symbol:

$$t^\sigma \neq u^\sigma$$

The axioms of PA^ω are given in Figure 3. $(refl)$ and $(Leib)$ are the axioms of equational theories, (Δs) , (Δk) , $(\Delta rec0)$ and $(\Delta recS)$ are the definitional axioms for the individual constants s , k and rec , and finally (Snz) and (ind) are the axiom stating that 0 is not a successor and the axiom scheme of induction for every formula A with free variables among $x^\iota, \vec{y}^{\vec{\sigma}}$. Since the predecessor function is definable in system T, the injectivity of the successor:

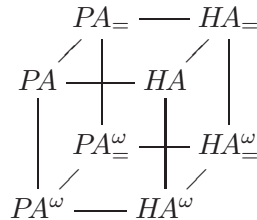
$$\forall x \forall y (Sx = Sy \Rightarrow x = y)$$

is derivable in PA^ω . Using the provable equality $\forall x (Skkx = x)$, we can prove in PA^ω :

$$\forall x \forall y (\forall z (zx = zy) \Rightarrow x = y)$$

but the status of $\forall x \forall y (\forall z (xz = yz) \Rightarrow x = y)$ is less clear and we won't consider it here.

1.3.1. Relation to usual theories. We discuss in this section about the relation between our version of PA^ω , in which the inequality predicate is atomic, and other intuitionistic or classical systems. First, in this section we will consider the following systems:



where the P/H part corresponds to Peano and Heyting arithmetic, the latter being the intuitionistic version of the former, the ω part corresponds to the possibility of having higher type variables, and the $=$ part corresponds to having an atomic predicate for equality while the absence of $=$ indicates a primitive inequality. The systems without higher type variables have addition and multiplication as individual constants and their defining equations as axioms, instead of s , k and rec and their defining equations. For the systems based on a primitive equality we refer to [TVD88]. In the case of primitive equality, the Leibniz scheme is:

$$\forall \vec{z}^{\vec{\tau}} \forall x^{\sigma} \forall y^{\sigma} (x = y \Rightarrow A \Rightarrow A \{y/x\})$$

while in the case of primitive inequality we define it is as in section 1.2:

$$\forall \vec{z}^{\vec{\tau}} \forall x^{\sigma} \forall y^{\sigma} (\neg A \Rightarrow A \{y/x\} \Rightarrow x \neq y)$$

Similarly, in $PA_{=}$ and $HA_{=}$ the injectivity of successor is:

$$\forall x^t \forall y^t (\mathbb{S} x = \mathbb{S} y \Rightarrow x = y)$$

which is derivable in $PA_{=}^{\omega}$ and $HA_{=}^{\omega}$, while in PA and HA we define it to be the following formula:

$$\forall x^t \forall y^t (x \neq y \Rightarrow \mathbb{S} x \neq \mathbb{S} y)$$

which is derivable in PA^{ω} and HA^{ω} .

It is well known that $HA_{=} \vdash \forall x^t \forall y^t (\neg \neg x = y \Rightarrow x = y)$, so in our system with only negative connectives, $HA_{=}$ and $PA_{=}$ prove exactly the same formulas. It is quite easy to prove also that $HA \vdash \forall x^t \forall y^t (\neg \neg x \neq y \Rightarrow x \neq y)$, so HA and PA also prove exactly the same formulas (again, in the particular case of our system with only negative connectives). However, despite the equivalence of provability between PA and HA , the associated proof terms are very different. Indeed, the proof of $HA \vdash \forall x^t \forall y^t (\neg \neg x \neq y \Rightarrow x \neq y)$ is obtained by a double induction on x and y , while in PA , a proof of $PA \vdash \forall x^t \forall y^t (\neg \neg x \neq y \Rightarrow x \neq y)$ can also be obtained through classical logic, in which case its computational content relies on the control features of languages for classical logic (like $\lambda\mu$ -calculus).

When we switch to higher type equalities, things become more complicated. Indeed, $HA_{=}^{\omega}$ fails to prove $\forall x^{\sigma} \forall y^{\sigma} (\neg \neg x = y \Rightarrow x = y)$, since we cannot perform induction on higher type variables, so the systems $HA_{=}^{\omega}$ and $PA_{=}^{\omega}$ are not equivalent anymore, even at the level of provability.

For classical systems PA^{ω} and $PA_{=}^{\omega}$, a formula with equalities and inequalities is provable in PA^{ω} (through the encoding $(x = y) \equiv (x \neq y \Rightarrow \perp)$) if and only if it is provable in $PA_{=}^{\omega}$ (through the encoding $(x \neq y) \equiv (x = y \Rightarrow \perp)$). However, for intuitionistic systems HA^{ω} and $HA_{=}^{\omega}$ things are less clear since double negations cannot be eliminated on atomic formulas. In particular, we did not investigate the relation between the system HA^{ω} with primitive inequalities and $HA_{=}^{\omega}$ or $PA^{\omega}/PA_{=}^{\omega}$.

1.4. The axiom of choice. In this section, A is a formula over $\Sigma_{PA^{\omega}}$ with free variables among $x^t, y^{\sigma}, z^{\sigma}, \vec{u}^{\vec{\tau}}$. For clarity we write $A \{t, u, v\}$ instead of $A \{t/x, u/y, v/z\}$. Classical analysis CA^{ω} (in the sense of [Koh08]) is defined to be PA^{ω} augmented with the following axiom scheme:

$$(DC) \quad \forall \vec{u}^{\vec{\tau}} (\forall x^t \forall y^{\sigma} \exists z^{\sigma} A \{x, y, z\} \Rightarrow \exists v^{\iota \rightarrow \sigma} \forall x^t A \{x, v x, v(\mathbb{S} x)\})$$

As proved in [Koh08], (DC) implies in PA^{ω} both countable choice:

$$\forall \vec{u}^{\vec{\tau}} (\forall x^t \exists y^{\sigma} B \Rightarrow \exists v^{\iota \rightarrow \sigma} \forall x^t B \{v x/y\})$$

where B has free variables among $x^t, y^\sigma, \vec{u}^{\vec{\tau}}$, and the more usual version of dependent choice:

$$\forall \vec{u}^{\vec{\tau}} (\forall x^t \forall y^\sigma \exists z^\sigma A \{x, y, z\} \Rightarrow \forall w^\sigma \exists v^{t \rightarrow \sigma} (v 0 = w \wedge \forall x^t A \{x, v x, v (S x)\}))$$

1.5. Models of classical multisorted first-order logic. In this section we recall what is a model of a given multisorted first-order theory. We fix a signature Σ .

Definition 1.3. A Σ -structure \mathcal{M} is given by:

- a set $\sigma^{\mathcal{M}}$ for each sort σ constructed from the base sorts of Σ
- an application function from $(\sigma \rightarrow \tau)^{\mathcal{M}} \times \sigma^{\mathcal{M}}$ to $\tau^{\mathcal{M}}$
- an element $c^{\mathcal{M}} \in \sigma^{\mathcal{M}}$ for each individual constant c^σ of Σ
- a set $P^{\mathcal{M}} \subseteq \sigma_1^{\mathcal{M}} \times \dots \times \sigma_n^{\mathcal{M}}$ for each sorted predicate $P(t_1^{\sigma_1}, \dots, t_n^{\sigma_n})$ of Σ

Using the application function, we can extend the interpretation to individuals with parameters: if t is an individual of sort σ with free variables $\vec{x}^{\vec{\tau}}$ and if \vec{a} are elements of $\vec{\tau}^{\mathcal{M}}$, then $(t \{\vec{a}/\vec{x}\})^{\mathcal{M}}$ is an element of $\sigma^{\mathcal{M}}$.

As usual in model theory, for any Σ -structure \mathcal{M} and any closed formula A on Σ we can define when \mathcal{M} validates A , written $\mathcal{M} \vDash A$. Now we fix a theory $\mathcal{A}x$ on Σ and we define what is a model of $\mathcal{A}x$:

Definition 1.4. If \mathcal{M} is a Σ -structure, then \mathcal{M} is a model of $\mathcal{A}x$ if for any $A \in \mathcal{A}x$:

$$\mathcal{M} \vDash A$$

The soundness theorem states that if \mathcal{M} is a model of $\mathcal{A}x$, then for any closed formula A , $\mathcal{A}x \vdash A$ implies $\mathcal{M} \vDash A$. In the particular case of equational theories we fix the interpretation of the \neq_σ predicate to be the following set of pairs of elements of $\sigma^{\mathcal{M}}$:

$$\neq_\sigma^{\mathcal{M}} \triangleq \{(a, b) \in \sigma^{\mathcal{M}} \times \sigma^{\mathcal{M}} \mid a \neq b\}$$

2. SYNTAX AND SEMANTICS OF $\lambda\mu$ -CALCULUS

In this section, we define the programming language to which we map classical proofs, and define the categorical model of this language. First, we describe $\lambda\mu$ -calculus as a typed language and give its equational theory under call-by-name semantics before defining how we map classical proofs in this language. Then we define the language μ PCF (that we will use in order to realize Peano arithmetic and the axiom of choice) as a $\lambda\mu$ -theory. Finally we describe categories of continuations as a model of call-by-name $\lambda\mu$ -calculus and we present the connection between the interpretation of $\lambda\mu$ -calculus in categories of continuations and that of its continuation-passing-style translation in the underlying cartesian category with exponentials of a fixed object.

2.1. $\lambda\mu$ -calculus. The $\lambda\mu$ -calculus is an extension of the λ -calculus introduced by Parigot in [Par92] in order to represent and evaluate classical proofs. In $\lambda\mu$ -calculus, there is another kind of variables along standard λ -variables: the μ -variables. Just like the λ -variables are bound by the construct $\lambda x.M$, the μ -variables (which will be written α, β, \dots) are bound by the new construct of $\lambda\mu$ -calculus: $\mu\alpha.M$. The other new construct, $[\alpha] M$ should be understood as a mean for M to get arguments which are passed to the enclosing $\mu\alpha.N$.

$\lambda\mu$ -calculus provides a direct interpretation of classical proofs just like λ -calculus is an interpretation of the intuitionistic ones. In classical sequent calculus, sequents are of the form $A_1, \dots, A_n \vdash B_1, \dots, B_m$, and should be interpreted as the formula $A_1 \wedge \dots \wedge A_n \Rightarrow B_1 \vee \dots \vee B_m$. The interpretation of a proof of such a sequent in $\lambda\mu$ -calculus is then a $\lambda\mu$ -term with free λ -variables x_1, \dots, x_n , and free μ -variables $\alpha_1, \dots, \alpha_m$, x_i representing the possibility to use the hypothesis A_i , and α_i representing the possibility to produce (part of) a proof of B_i . Under this view, a $\lambda\mu$ -term may provide many proofs at the same time, each proof being given step-by-step.

In the original version of [Par92], the terms were restricted to λ -abstractions, applications and $\mu\alpha. [\beta] M$ for some term M . This syntax was extended in [dG94] by allowing terms $\mu\alpha.M$ and $[\alpha] M$. This extension of syntax was used in [Sau05] together with a well-chosen set of reduction rules (this calculus being called $\Lambda\mu$ -calculus) to recover the separation property that failed in Parigot's $\lambda\mu$ -calculus, as shown in [DP01]. Later on in [AHS07], Parigot's version was related to minimal classical logic, and De Groote's to full classical logic. Here we use the version of [Sel01] (but without disjunction types), which is in the lines of [dG94, Sau05]. We recommend [HS09] for an historical overview of the different versions of $\lambda\mu$ -calculus and how they relate to each other.

2.1.1. Type system. The types of $\lambda\mu$ -calculus are those of simply typed λ -calculus with a fixed set of base types (ranged over by X) together with a product type and a distinguished empty type 0 used to type terms $[\alpha] M$:

$$T, U ::= X \mid T \rightarrow U \mid T \times U \mid 0$$

In addition to the base types, $\lambda\mu$ -calculus is parameterized with a fixed set of typed constants: $\mathcal{Cst} = \{c : T, \dots\}$. These two parameters form a signature of $\lambda\mu$ -calculus:

Definition 2.1 ($\lambda\mu$ signature). A $\lambda\mu$ signature is a set of base types together with a set of typed constants.

From a $\lambda\mu$ signature, we will now define the derivable typing judgments, which are presented as sequents of the form:

$$x_1 : T_1, \dots, x_n : T_n \vdash M : T \mid \alpha_1 : U_1, \dots, \alpha_m : U_m$$

where the free λ -variables of M are among x_1, \dots, x_n and its free μ -variables are among $\alpha_1, \dots, \alpha_m$. The $x_1 : T_1, \dots, x_n : T_n$ part will be called the λ -context, and $\alpha_1 : U_1, \dots, \alpha_m : U_m$ the μ -context. If the μ -context is empty, then we will simply write:

$$x_1 : T_1, \dots, x_n : T_n \vdash M : T$$

The derivable typing judgments are defined in Figure 4 for a given signature of $\lambda\mu$ -calculus. The left and right weakening rules are admissible in that type system, and we use them

$$\begin{array}{c}
 \frac{}{\vec{x} : \vec{T}, x : T \vdash x : T \mid \vec{\alpha} : \vec{U}} \quad \frac{}{\vec{x} : \vec{T} \vdash c : T \mid \vec{\alpha} : \vec{U}} \text{ (c:T} \in \mathcal{Cst}\text{)} \\
 \frac{\vec{x} : \vec{T}, x : T \vdash M : U \mid \vec{\alpha} : \vec{U}}{\vec{x} : \vec{T} \vdash \lambda x.M : T \rightarrow U \mid \vec{\alpha} : \vec{U}} \quad \frac{\vec{x} : \vec{T} \vdash M : T \rightarrow U \mid \vec{\alpha} : \vec{U} \quad \vec{x} : \vec{T} \vdash N : T \mid \vec{\alpha} : \vec{U}}{\vec{x} : \vec{T} \vdash MN : U \mid \vec{\alpha} : \vec{U}} \\
 \frac{\vec{x} : \vec{T} \vdash M : T \mid \vec{\alpha} : \vec{U} \quad \vec{x} : \vec{T} \vdash N : U \mid \vec{\alpha} : \vec{U}}{\vec{x} : \vec{T} \vdash \langle M, N \rangle : T \times U \mid \vec{\alpha} : \vec{U}} \quad \frac{\vec{x} : \vec{T} \vdash M : T_1 \times T_2 \mid \vec{\alpha} : \vec{U}}{\vec{x} : \vec{T} \vdash \pi_i M : T_i \mid \vec{\alpha} : \vec{U}} \\
 \frac{\vec{x} : \vec{T} \vdash M : U \mid \alpha : U, \vec{\alpha} : \vec{U}}{\vec{x} : \vec{T} \vdash [\alpha]M : 0 \mid \alpha : U, \vec{\alpha} : \vec{U}} \quad \frac{\vec{x} : \vec{T} \vdash M : 0 \mid \alpha : U, \vec{\alpha} : \vec{U}}{\vec{x} : \vec{T} \vdash \mu\alpha.M : U \mid \vec{\alpha} : \vec{U}}
 \end{array}$$

 Figure 4: Typing rules of $\lambda\mu$ -calculus

without explicitly mentioning it. Here are some examples of derivable typing judgments.

$$\begin{array}{c}
 \vdash \lambda x.\mu\alpha.x : 0 \rightarrow T \quad \vdash \lambda y.\mu\alpha.y (\lambda x.[\alpha]x) : ((T \rightarrow 0) \rightarrow 0) \rightarrow T \\
 \vdash \lambda y.\mu\alpha.[\alpha]y (\lambda x.\mu\beta.[\alpha]x) : ((T \rightarrow U) \rightarrow T) \rightarrow T
 \end{array}$$

The logical counterparts of these terms are respectively the ex falso formula $\perp \Rightarrow A$, the double-negation elimination $((A \Rightarrow \perp) \Rightarrow \perp) \Rightarrow A$ and the law of Peirce $((A \Rightarrow B) \Rightarrow A) \Rightarrow A$, these last two being classical principles. Remark that in the first judgment we write $\mu\alpha.x$, and in the second example we write $\lambda x.[\alpha]x$. This is a consequence of using the extended syntax of [dG94, Sau05], where the empty type is handled as any other. The third term corresponds to the `call/cc` operator of the Scheme programming language which, as observed by Griffin in [Gri90], can be typed with the law of Peirce.

2.1.2. Interpreting classical logic in $\lambda\mu$ -calculus. Here we interpret every formula A as a type A^* of $\lambda\mu$ -calculus, and every proof of a sequent $\Gamma \vdash A \mid \Delta$ in a given theory as a typing derivation of a term $\Gamma^* \vdash M : A^* \mid \Delta^*$. The interpretation of the axioms is for the time being a parameter of the interpretation of proofs, which is the first component of our realizability interpretation of classical logic.

Fix a first-order signature Σ and a $\lambda\mu$ signature. Fix also for each predicate P of Σ a type P^* of $\lambda\mu$ -calculus. We extend this interpretation to every formula over Σ the following way:

$$\perp^* = 0 \quad (A \Rightarrow B)^* = A^* \rightarrow B^* \quad (A \wedge B)^* = A^* \times B^* \quad (\forall x^\sigma A)^* = A^*$$

An important fact is that during this interpretation we simply forget about first-order quantifications. This is a Curry-style interpretation (as opposed to a Church-style interpretation). The effect of this is that the term interpreting a proof of a universal statement must not depend on the particular individual. This is indeed true for the equational axioms, but when it comes to the axiom schemes of induction and choice it is no longer the case. For that reason we will introduce in section 3.1 a relativization predicate. The interpretation of a formula as a type of $\lambda\mu$ -calculus is then extended to contexts: Γ^* (resp. Δ^*) is a context of λ -variables (resp. μ -variables) with types B^* for B in Γ (resp. Δ).

Fix now a first-order theory $\mathcal{A}x$ and for each $A \in \mathcal{A}x$ a closed $\lambda\mu$ -term M_A of type A^* which is a parameter of the interpretation. The interpretation of first-order proofs as

$$\begin{aligned}
\left(\frac{}{\Gamma, A \vdash A \mid \Delta} \right)^* &= \overline{\Gamma^*, x : A^* \vdash x : A^* \mid \Delta^*} \\
\left(\frac{}{\Gamma \vdash A \mid \Delta} \right)^{*(A \in \mathcal{A}x)} &= \Gamma^* \vdash M_A : A^* \mid \Delta^* \\
\left(\frac{\Gamma, A \vdash B \mid \Delta}{\Gamma \vdash A \Rightarrow B \mid \Delta} \right)^* &= \frac{\Gamma^*, x : A^* \vdash M : B^* \mid \Delta^*}{\Gamma^* \vdash \lambda x. M : A^* \rightarrow B^* \mid \Delta^*} \\
\left(\frac{\Gamma \vdash A \Rightarrow B \mid \Delta \quad \Gamma \vdash A \mid \Delta}{\Gamma \vdash B \mid \Delta} \right)^* &= \frac{\Gamma^* \vdash M : A^* \rightarrow B^* \mid \Delta^* \quad \Gamma^* \vdash N : A^* \mid \Delta^*}{\Gamma^* \vdash MN : B^* \mid \Delta^*} \\
\left(\frac{\Gamma \vdash A \mid \Delta \quad \Gamma \vdash B \mid \Delta}{\Gamma \vdash A \wedge B \mid \Delta} \right)^* &= \frac{\Gamma^* \vdash M : A^* \mid \Delta^* \quad \Gamma^* \vdash N : B^* \mid \Delta^*}{\Gamma^* \vdash \langle M, N \rangle : A^* \times B^* \mid \Delta^*} \\
\left(\frac{\Gamma \vdash A_1 \wedge A_2 \mid \Delta}{\Gamma \vdash A_i \mid \Delta} \right)^* &= \frac{\Gamma^* \vdash M : A_1^* \times A_2^* \mid \Delta^*}{\Gamma^* \vdash \pi_i M : A_i^* \mid \Delta^*} \\
\left(\frac{\Gamma \vdash A \mid \Delta}{\Gamma \vdash \forall x^\sigma A \mid \Delta} \right)^{*(x^\sigma \notin \text{FV}(\Gamma, \Delta))} &= \Gamma^* \vdash M : A^* \mid \Delta^* \\
\left(\frac{\Gamma \vdash \forall x^\sigma A \mid \Delta}{\Gamma \vdash A \{t^\sigma/x^\sigma\} \mid \Delta} \right)^* &= \Gamma^* \vdash M : A^* \mid \Delta^* \\
\left(\frac{\Gamma \vdash A \mid A, \Delta}{\Gamma \vdash \perp \mid A, \Delta} \right)^* &= \frac{\Gamma^* \vdash M : A^* \mid \alpha : A^*, \Delta^*}{\Gamma^* \vdash [\alpha]M : 0 \mid \alpha : A^*, \Delta^*} \\
\left(\frac{\Gamma \vdash \perp \mid A, \Delta}{\Gamma \vdash A \mid \Delta} \right)^* &= \frac{\Gamma^* \vdash M : 0 \mid \alpha : A^*, \Delta^*}{\Gamma^* \vdash \mu\alpha. M : A^* \mid \Delta^*}
\end{aligned}$$

Figure 5: Interpretation of classical proofs in $\lambda\mu$ -calculus

typing derivations in $\lambda\mu$ -calculus is given in Figure 5. In this interpretation, the structural rules on the right part of a sequent are interpreted with the μ constructs of $\lambda\mu$ -calculus. Indeed, these rules are the ones which make our system a classical one, so it doesn't come as a surprise that they are interpreted using the control features of $\lambda\mu$ -calculus.

2.1.3. $\lambda\mu$ theories. We follow [Sel01] and define $\lambda\mu$ -calculus as an equational theory. We only consider here the case of call-by-name semantics. The axioms of the call-by-name $\lambda\mu$ -calculus are given in Figure 6, where in each equation the two terms are typed with the same type. In the equations (ζ_{\rightarrow}) , (ζ_{\times}) and (ζ_0) , $_$ is a placeholder for the term coming after $[\alpha]$, i.e. $M \{[\alpha] _ N / [\alpha] _ \}$ is obtained by replacing in M all the subterms of the form $[\alpha] P$ with $[\alpha] P N$. From these axioms we define the notion of $\lambda\mu$ theory:

$$\begin{array}{ll}
 (\beta_{\rightarrow}) & (\lambda x.M) N = M \{N/x\} & (\eta_{\rightarrow}) & \lambda x.M x = M & (x \notin \text{FV}(M)) \\
 (\beta_{\times}) & \pi_i \langle M_1, M_2 \rangle = M_i & (\eta_{\times}) & \langle \pi_1 M, \pi_2 M \rangle = M \\
 (\beta_0) & [\alpha] \mu\beta.M = M \{\alpha/\beta\} & (\eta_0) & \mu\alpha. [\alpha] M = M & (\alpha \notin \text{FV}(M)) \\
 \\
 (\zeta_{\rightarrow}) & (\mu\alpha.M) N = \mu\alpha.M \{[\alpha] _ N / [\alpha] _ \} \\
 (\zeta_{\times}) & \pi_i (\mu\alpha.M) = \mu\alpha.M \{[\alpha] \pi_i _ / [\alpha] _ \} \\
 (\zeta_0) & \mu\alpha.M = M \{ _ / [\alpha] _ \} & & & (\alpha : 0)
 \end{array}$$

 Figure 6: Axioms of the call-by-name $\lambda\mu$ -calculus

Definition 2.2 ($\lambda\mu$ theory). A $\lambda\mu$ theory is a set of equations between typed terms of the same type (with free variables of the same type) which contains the axioms of call-by-name $\lambda\mu$ -calculus and is a congruence (contextually-closed equivalence relation).

We use here a slightly different set of axioms from that of Selinger [Sel01]. Our β_0 and η_0 equations are the β_μ and η_μ equations of Selinger, and we replace his β_\perp (which is $[\alpha] M = M$ if $M : 0$) with our ζ_0 . However, the two systems are equivalent:

Lemma 2.3. *Under the contextual closures of β_0 and η_0 (the β_μ and η_μ of Selinger), the equation β_\perp of Selinger is equivalent to ζ_0 .*

Proof. Suppose β_\perp holds, let $M : 0$ and α be a μ -variable of type 0. Using the contextual closure of β_\perp , we have $M = M \{ _ / [\alpha] _ \}$, and again by β_\perp we get $M = [\alpha] M \{ _ / [\alpha] _ \}$. Then by contextual closure $\mu\alpha.M = \mu\alpha. [\alpha] M \{ _ / [\alpha] _ \}$, which is equal to $M \{ _ / [\alpha] _ \}$ using η_0 , since α does not appear free in $M \{ _ / [\alpha] _ \}$.

Conversely, suppose ζ_0 holds, let $M : 0$, α be a μ -variable of type 0 and β be a μ -variable of type 0 which does not appear free in M . Using ζ_0 we have $M = \mu\beta.M$, so by contextual closure $[\alpha] M = [\alpha] \mu\beta.M$, so by β_0 we get $[\alpha] M = M \{\alpha/\beta\}$, but $M \{\alpha/\beta\} = M$ since β does not appear free in M , and we get finally $[\alpha] M = M$. \square

Replacing Selinger's β_\perp allows us to have a more symmetric calculus, with a set of three equations for each type constructor, the ζ equations representing the transmission of the context of a $\mu\alpha.M$ to the subterms $[\alpha] N$.

2.1.4. μ PCF. We give here an example of a $\lambda\mu$ theory called μ PCF, which will also be the language in which we will interpret Peano arithmetic and the axiom of choice. Programming language for Computable Functions (PCF) is a functional programming language described by Plotkin in [Pl77]. It is based on Scott's Logic for Computable Functions (LCF), which was presented in [Sco93]. The language contains constants for natural numbers and general recursion. It is probably the simplest example of a Turing-complete higher-order language. Here we consider an extension of PCF to primitively handle control operators, by presenting PCF as a $\lambda\mu$ theory. In [OS97] the authors define a call-by-value semantics for $\lambda\mu$ -calculus, and they illustrate it with μ PCF_V, a call-by-value version of PCF with control. Later on, Laird defines in [Lai99] its call-by-name version, which is the version we use here. The choice of this language is justified by our will to get computational content directly from classical proofs.

$$\begin{array}{lll}
\Upsilon M = M(\Upsilon M) & \text{succ } \bar{n} = \overline{n+1} & \text{succ } (\mu\alpha.M) = M \{ \text{succ } (_) / [\alpha] _ \} \\
\text{pred } \bar{0} = \bar{0} & \text{pred } \overline{n+1} = \bar{n} & \text{pred } (\mu\alpha.M) = M \{ \text{pred } (_) / [\alpha] _ \} \\
\text{if}_0 \bar{0} M N = M & \text{if}_0 \overline{n+1} M N = N & \text{if}_0 (\mu\alpha.M) N P = M \{ \text{if}_0 (_) N P / [\alpha] _ \}
\end{array}$$

Figure 7: Equations of μ PCF

Our version of μ PCF has only one base type for natural numbers: I , and the constants are:

$$\bar{n} : I \quad \text{succ} : I \rightarrow I \quad \text{pred} : I \rightarrow I \quad \text{if}_0 : I \rightarrow T \rightarrow T \quad \Upsilon : (T \rightarrow T) \rightarrow T$$

It will also be useful to have a canonical term on each type so we define:

$$\Omega \triangleq \Upsilon (\lambda x. x) : T$$

This term represents non-termination, or “undefined”. The equations of μ PCF given in Figure 7 are standard and include the interactions of the constants with the μ operator. We can finally define μ PCF as the $\lambda\mu$ theory generated by these equations:

Definition 2.4 (μ PCF). The $\lambda\mu$ theory μ PCF is the smallest $\lambda\mu$ theory containing the equations of Figure 7.

2.2. Categories of continuations. Categories of continuations are to call-by-name $\lambda\mu$ -calculus what cartesian closed categories are to λ -calculus, in the sense that if we fix a signature, there is a one-to-one correspondence between $\lambda\mu$ theories and categories of continuations together with an interpretation of the signature. Ong defines $\lambda\mu$ categories in [Ong96] by reformulating the syntax of $\lambda\mu$ -calculus in categorical terms. Later on, Hofmann and Streicher proved the soundness and completeness of categories of continuations with respect to $\lambda\mu$ -calculus, providing the first abstract version of $\lambda\mu$ -categories. Finally, Selinger axiomatized these in [Sel01] under the name of control categories, proving that they are equivalent (for a suitable notion of equivalence based on weak functors) to categories of continuations, and therefore sound and complete with respect to call-by-name $\lambda\mu$ -calculus. Moreover, he proved that the categorical dual of control categories are sound and complete with respect to call-by-value $\lambda\mu$ -calculus. Here we are only interested in the call-by-name version, and we use the model of categories of continuations.

Definition 2.5 (Category of continuations). Let \mathcal{C} be a distributive category, that is, a category with finite products and coproducts such that the canonical distributivity morphisms from $A \times B + A \times C$ to $A \times (B + C)$ are isomorphisms (which implies that the morphism from $\mathbf{0}$ to $A \times \mathbf{0}$ is also an isomorphism), and let $R \in \text{Ob}(\mathcal{C})$ be a fixed object such that all exponentials R^A for $A \in \text{Ob}(\mathcal{C})$ exist. Then the full subcategory $R^{\mathcal{C}}$ of \mathcal{C} consisting of the objects R^A for $A \in \mathcal{C}$ is called a category of continuations.

We will differentiate morphisms in \mathcal{C} and $R^{\mathcal{C}}$ by writing \mathfrak{K} , \mathfrak{K}^1 for morphisms in \mathcal{C} and ϕ , ψ , ζ , ξ for morphisms in $R^{\mathcal{C}}$. As observed in [LRS93], a category of continuations $R^{\mathcal{C}}$ is in particular a cartesian closed category:

¹Pronounced as in Доктор Живаго (Doctor Zhivago).

Lemma 2.6. *If R^C is a category of continuations, then R^0 defines a terminal object and R^{A+B} defines a cartesian product of R^A and R^B , so R^C is cartesian. Moreover, if $A \in \text{Ob}(\mathcal{C})$ and $R^B \in \text{Ob}(R^C)$, then $R^{A \times B}$ defines an exponential in \mathcal{C} of R^B by A . Consequently, R^C is cartesian closed, the exponential of R^B by R^A being $R^{R^A \times B}$.*

Be careful that we have two (isomorphic) terminal objects, one in \mathcal{C} and one in R^C , and two (isomorphic) products of R^A and R^B , again one in \mathcal{C} and one in R^C . To avoid confusion and without loss of generality we will suppose that they are equal: $R^0 = \mathbf{1}$ and $R^{A+B} = R^A \times R^B$. For the same reason, we also suppose $R = R^1$.

2.2.1. *Classical disjunction in categories of continuations.* We could have added a primitive connective \vee for the disjunction in the logic, with the following rules:

$$\frac{\Gamma \vdash A \vee B \mid \Delta, A, B}{\Gamma \vdash \perp \mid \Delta, A, B} \qquad \frac{\Gamma \vdash \perp \mid \Delta, A, B}{\Gamma \vdash A \vee B \mid \Delta}$$

and then interpret these logical rules by adding the following typing rules to $\lambda\mu$ -calculus:

$$\frac{\vec{x} : \vec{T} \vdash M : U_1 \wp U_2 \mid \vec{\alpha} : \vec{U}, \beta_1 : U_1, \beta_2 : U_2}{\vec{x} : \vec{T} \vdash [\beta_1, \beta_2] M : 0 \mid \vec{\alpha} : \vec{U}, \beta_1 : U_1, \beta_2 : U_2} \qquad \frac{\vec{x} : \vec{T} \vdash M : 0 \mid \vec{\alpha} : \vec{U}, \beta_1 : U_1, \beta_2 : U_2}{\vec{x} : \vec{T} \vdash \mu(\beta_1, \beta_2).M : U_1 \wp U_2 \mid \vec{\alpha} : \vec{U}}$$

These rules are present in [Sel01], however we choose here to keep things simple and stick to the usual $\lambda\mu$ -calculus without disjunction types. Nevertheless we still use the binoidal functor \wp in categories of continuations to interpret multi-conclusioned sequents. Following [Sel01] we write: $R^A \wp R^B \triangleq R^{A \times B}$ for the interpretation of the classical disjunction between R^A and R^B , $i : R \rightarrow R^A$ for the interpretation of the right weakening rule and $\nabla : R^A \wp R^A \rightarrow R^A$ for the interpretation of the right contraction rule.

Another interesting fact about categories of continuations is that we can define a functor from \mathcal{C}^{op} to R^C which maps $A \in \text{Ob}(\mathcal{C})$ to $R^A \in \text{Ob}(R^C)$, and $\varkappa : A \rightarrow B$ to $R^\varkappa : R^B \rightarrow R^A$ which is the currying of:

$$R^B \times A \xrightarrow{\text{Id}_{R^B} \times \varkappa} R^B \times B \xrightarrow{\text{ev}} R$$

The morphism $i : R \rightarrow R^A$ corresponds to $R^{\mathbf{1}_A}$ (where $\mathbf{1}_A$ is the unique morphism from A to the terminal object $\mathbf{1}$) and the morphism $\nabla : R^A \wp R^A \rightarrow R^A$ corresponds to $R^{\text{pair}(\text{Id}_A, \text{Id}_A)}$ (remember that $R^A \wp R^A = R^{A \times A}$). Also, in particular, if $A \simeq B$ in \mathcal{C} , then $R^A \simeq R^B$ in R^C (and in \mathcal{C}). Through this functor, the cocartesian structure of \mathcal{C} translates to the cartesian structure of R^C .

2.2.2. *Interpretation of $\lambda\mu$ -calculus in categories of continuations.* We describe here the interpretation of call-by-name $\lambda\mu$ -calculus in a category of continuations as defined in [Sel01]. We fix a signature of $\lambda\mu$ -calculus and a category of continuations R^C . To each type T of $\lambda\mu$ -calculus we associate an object $\llbracket T \rrbracket \in \text{Ob}(\mathcal{C})$ of continuations of type T , and an object $[T] = R^{\llbracket T \rrbracket} \in \text{Ob}(R^C)$ of computations of type T . The objects $\llbracket X \rrbracket \in \text{Ob}(\mathcal{C})$ where X is a base type of the signature are parameters of the interpretation, and we define inductively:

$$\begin{aligned} \llbracket T \rightarrow U \rrbracket &= [T] \times \llbracket U \rrbracket \in \text{Ob}(\mathcal{C}) & \llbracket T \times U \rrbracket &= \llbracket T \rrbracket + \llbracket U \rrbracket \in \text{Ob}(\mathcal{C}) \\ \llbracket 0 \rrbracket &= \mathbf{1} & [T] &= R^{\llbracket T \rrbracket} \in \text{Ob}(R^C) \end{aligned}$$

We have in particular $[T \times U] = R^{[T]+[U]} = R^{[T]} \times R^{[U]} = [T] \times [U]$ where \times is the cartesian product in $R^{\mathcal{C}}$ defined above, $[T \rightarrow U] = R^{[T] \times [U]} = R^{R^{[T]} \times [U]} = (R^{[U]})^{R^{[T]}} = [U]^{[T]}$, using the definition of the exponential in $R^{\mathcal{C}}$ given above, and $[0] = R^{\mathbf{1}} = R$.

Once we have an interpretation of types in $R^{\mathcal{C}}$, we define the interpretation of typed $\lambda\mu$ -terms such that a term:

$$x_1 : T_1, \dots, x_n : T_n \vdash M : V \mid \alpha_1 : U_1, \dots, \alpha_m : U_m$$

is interpreted as a morphism in $R^{\mathcal{C}}$:

$$[x_1 : T_1, \dots, x_n : T_n \vdash M : V \mid \alpha_1 : U_1, \dots, \alpha_m : U_m] : [T_1] \times \dots \times [T_n] \rightarrow [V] \wp [U_1] \wp \dots \wp [U_m]$$

where $[T_1] \times \dots \times [T_n]$ associates to the left and $[V] \wp [U_1] \wp \dots \wp [U_m]$ associates to the right. In order to do that, we suppose given for each constant $c : T \in \mathcal{C}st$ of the signature a morphism in $R^{\mathcal{C}}$:

$$[c : T] : \mathbf{1} \rightarrow [T]$$

which is again a parameter of the interpretation. These parameters are summarized in the following definition:

Definition 2.7 (Interpretation). Given a signature and a category of continuations $R^{\mathcal{C}}$, an interpretation of $\lambda\mu$ -calculus is given by an object $\llbracket X \rrbracket \in \mathcal{C}$ for each base type X of the signature and a morphism $[c : T] : \mathbf{1} \rightarrow [T]$ in $R^{\mathcal{C}}$ for each constant $c : T \in \mathcal{C}st$ of the signature.

We now have all necessary material to interpret every typed $\lambda\mu$ -term as a morphism in $R^{\mathcal{C}}$. The interpretation of typed $\lambda\mu$ -terms is almost identical to the interpretation of λ -calculus in a cartesian closed category (since as shown in the previous section, $R^{\mathcal{C}}$ is cartesian closed). The first difference is that we must be able to carry over the μ -context, so we want to build from $\phi : [T] \rightarrow [U]$ a morphism $\phi \wp [V] : [T] \wp [V] \rightarrow [U] \wp [V]$. The second difference is that in order to interpret the introduction rules for $\mu\alpha.M$ and $[\alpha] M$, we also need to have canonical morphisms from $[0] \wp [\vec{U}] \wp [T]$ to $[T] \wp [\vec{U}]$ and from $[T] \wp [\vec{U}] \wp [T]$ to $[0] \wp [\vec{U}] \wp [T]$. These requirements are axiomatized in [Sel01], to which we refer for the full definition of the interpretation, and the proof that the axioms of call-by-name $\lambda\mu$ -calculus are sound under this interpretation.

A model of a $\lambda\mu$ theory is then a sound interpretation:

Definition 2.8 (Model of a $\lambda\mu$ theory). A model of a $\lambda\mu$ theory is a category of continuations together with an interpretation for the $\lambda\mu$ signature such that every equation $M = N$ of the theory is true in the model: $\llbracket M \rrbracket = \llbracket N \rrbracket$.

If the $\lambda\mu$ theory is generated from a given set of equations, then any interpretation satisfying these equations is a model of the $\lambda\mu$ theory.

Since call-by-name $\lambda\mu$ -calculus is the internal language of categories of continuations (as shown in [Sel01]), we can apply $\lambda\mu$ -calculus constructs on morphisms of $R^{\mathcal{C}}$ through the use of $\lambda\mu$ -terms with parameters in $R^{\mathcal{C}}$. Therefore, we will also drop the interpretation brackets for terms. For example, if $\phi : \prod_{j \in J} R^{A_j} \rightarrow R^B \wp (\wp_{k \in K} R^{D_k})$ and $\psi : \prod_{j \in J} R^{A_j} \rightarrow R^{B \times C} \wp (\wp_{k \in K} R^{D_k})$, then $\psi \phi : \prod_{j \in J} R^{A_j} \rightarrow R^C \wp (\wp_{k \in K} R^{D_k})$, where formally $\psi \phi$ is the term with parameters $(y x) \{\phi/x, \psi/y\}$.

$$\begin{array}{c}
 \frac{}{\vec{k} : \vec{T}, k : T \vdash k : T} \quad \frac{}{\vec{k} : \vec{T} \vdash \tilde{c} : (\vec{T} \rightarrow R)} \quad (\tilde{c} : (\vec{T} \rightarrow R) \in \mathcal{Cst}) \\
 \\
 \frac{\vec{k} : \vec{T}, k : T \vdash M : R}{\vec{k} : \vec{T} \vdash \lambda k.M : T \rightarrow R} \quad \frac{\vec{k} : \vec{T} \vdash M : T \rightarrow R \quad \vec{k} : \vec{T} \vdash N : T}{\vec{k} : \vec{T} \vdash MN : R} \\
 \\
 \frac{}{\Gamma \vdash * : 1} \quad \frac{\vec{k} : \vec{T} \vdash M : T \quad \vec{k} : \vec{T} \vdash N : U}{\vec{k} : \vec{T} \vdash \langle M, N \rangle : T \times U} \quad \frac{\vec{k} : \vec{T} \vdash M : T_1 \times T_2}{\vec{k} : \vec{T} \vdash \pi_i M : T_i} \\
 \\
 \frac{\Gamma \vdash M : T_i}{\Gamma \vdash \text{in}_i M : T_1 + T_2} \quad \frac{\Gamma \vdash M : T_1 + T_2 \quad \Gamma, k : T_1 \vdash N_1 : U \quad \Gamma, k : T_2 \vdash N_2 : U}{\Gamma \vdash \text{case } M \{ \text{in}_1 k \mapsto N_1 \mid \text{in}_2 k \mapsto N_2 \} : U}
 \end{array}$$

 Figure 8: Typing rules of $\lambda^{R \times +}$

2.2.3. *Connection with the call-by-name CPS translation of $\lambda\mu$ -calculus.* Another interesting thing about the interpretation of $\lambda\mu$ -calculus into a category of continuations is the exact correspondence with the interpretation of its call-by-name CPS translation in the underlying cartesian “ R -closed” category, as stressed in [HS02]. The target of such a translation is a simply-typed λ -calculus $\lambda^{R \times +}$ with product and sum types, a particular base type R , and the function types being restricted to $T \rightarrow R$. This particular λ -calculus can be interpreted in \mathcal{C} by interpreting the product type as the product in \mathcal{C} , the sum type as the coproduct, and using the fact that function types are of the form $T \rightarrow R$, so having all exponentials R^A in \mathcal{C} is enough.

To be more precise, $\lambda^{R \times +}$ has one base type \tilde{X} for each base type X of $\lambda\mu$ -calculus and another particular base type R . From these we build the types:

$$T, U ::= \tilde{X} \mid T \rightarrow R \mid T \times U \mid 1 \mid T + U$$

The arrow types are syntactically restricted to be of the form $T \rightarrow R$. We map every type T of $\lambda\mu$ -calculus to a type \tilde{T} of $\lambda^{R \times +}$ (each base type X being obviously mapped to \tilde{X}) as follows:

$$\widetilde{T \rightarrow U} = (\tilde{T} \rightarrow R) \times \tilde{U} \quad \widetilde{T \times U} = \tilde{T} \times \tilde{U} \quad \tilde{0} = 1$$

$\lambda^{R \times +}$ also has one constant $\tilde{c} : \tilde{T} \rightarrow R$ for each constant $c : T$ of the source language. We also suppose given for each λ -variable $x : T$ of the source language a variable $\tilde{x} : \tilde{T} \rightarrow R$ in the target language, and for each μ -variable $\alpha : U$ of the source language a variable $\tilde{\alpha} : \tilde{U}$ in the target language. A typed $\lambda\mu$ -term:

$$x_1 : T_1, \dots, x_n : T_n \vdash M : T \mid \alpha_1 : U_1, \dots, \alpha_m : U_m$$

will then be translated to a typed λ -term:

$$\tilde{x}_1 : \tilde{T}_1 \rightarrow R, \dots, \tilde{x}_n : \tilde{T}_n \rightarrow R, \tilde{\alpha}_1 : \tilde{U}_1, \dots, \tilde{\alpha}_m : \tilde{U}_m \vdash \tilde{M} : \tilde{T} \rightarrow R \quad (2.1)$$

Before defining the translation, we give the typing rules of $\lambda^{R \times +}$ in Figure 8. The typing rules for the arrow type are restricted to the case $T \rightarrow R$. The translation of a variable x or a constant c is \tilde{x} or \tilde{c} as defined above, and the remaining part is given in Figure 9, where k is always a fresh variable. We define the interpretation of $\lambda^{R \times +}$ in \mathcal{C} by first giving an object $\llbracket T \rrbracket$ of \mathcal{C} for each type T :

$$\llbracket \tilde{X} \rrbracket = \llbracket X \rrbracket \quad \llbracket T \rightarrow R \rrbracket = R^{\llbracket T \rrbracket} \quad \llbracket T \times U \rrbracket = \llbracket T \rrbracket \times \llbracket U \rrbracket \quad \llbracket 1 \rrbracket = \mathbf{1} \quad \llbracket T + U \rrbracket = \llbracket T \rrbracket + \llbracket U \rrbracket$$

$$\begin{array}{ll}
\widetilde{\lambda x.M} = \lambda k. (\lambda \widetilde{x.M}) (\pi_1 k) (\pi_2 k) & \widetilde{M N} = \lambda k. \widetilde{M} \langle \widetilde{N}, k \rangle \\
\widetilde{\langle M, N \rangle} = \lambda k. \text{case } k \{ \text{in}_1 l \mapsto \widetilde{M} l \mid \text{in}_2 l \mapsto \widetilde{N} l \} & \widetilde{\pi_i M} = \lambda k. \widetilde{M} \text{in}_i k \\
\widetilde{\mu \alpha.M} = \lambda \widetilde{\alpha}. \widetilde{M} * & \widetilde{[\alpha] M} = \lambda k. \widetilde{M} \widetilde{\alpha}
\end{array}$$

Figure 9: Translation of $\lambda\mu$ -calculus in $\lambda^{R \times +}$

$$\begin{array}{ll}
(\beta_{\rightarrow}^{\lambda}) \quad (\lambda k.M) N = M \{N/k\} & (\eta_{\rightarrow}^{\lambda}) \quad \lambda k.M k = M \quad (k \notin \text{FV}(M)) \\
(\beta_{\times}^{\lambda}) \quad \pi_i \langle M_1, M_2 \rangle = M_i & (\eta_{\times}^{\lambda}) \quad \langle \pi_1 M, \pi_2 M \rangle = M \\
(\beta_{+}^{\lambda}) \quad \text{case } (\text{in}_i M) \left\{ \begin{array}{l} \text{in}_1 k \mapsto N_1 \\ \text{in}_2 k \mapsto N_2 \end{array} \right\} = N_i \{M/k\} & (\eta_{+}^{\lambda}) \quad \text{case } M \left\{ \begin{array}{l} \text{in}_1 k \mapsto \text{in}_1 k \\ \text{in}_2 k \mapsto \text{in}_2 k \end{array} \right\} = M \\
& (\eta_{!}^{\lambda}) \quad * = M
\end{array}$$

Figure 10: Axioms of $\lambda^{R \times +}$

The notation $\llbracket _ \rrbracket$ may seem misleading, but it is on purpose, since one can easily see that if T is a type of $\lambda\mu$ -calculus, then $\llbracket \widetilde{T} \rrbracket = \llbracket T \rrbracket$, where on the left the type \widetilde{T} of $\lambda^{R \times +}$ is interpreted in \mathcal{C} , and on the right the type T of $\lambda\mu$ -calculus is interpreted in $R^{\mathcal{C}}$. Since the only function types of our λ -calculus are of the form $T \rightarrow R$ and since \mathcal{C} has all exponentials R^A , we can interpret \widetilde{M} in \mathcal{C} the same way we would interpret simply typed λ -calculus in a cartesian closed category. The term \widetilde{M} of (2.1) is interpreted as:

$$\widetilde{M} : R^{\llbracket \widetilde{T}_1 \rrbracket} \times \dots \times R^{\llbracket \widetilde{T}_n \rrbracket} \times \llbracket \widetilde{U}_1 \rrbracket \times \dots \times \llbracket \widetilde{U}_m \rrbracket \rightarrow R^{\llbracket \widetilde{T} \rrbracket}$$

which is, by the above observation that $\llbracket \widetilde{T} \rrbracket = \llbracket T \rrbracket$:

$$\widetilde{M} : R^{\llbracket T_1 \rrbracket} \times \dots \times R^{\llbracket T_n \rrbracket} \times \llbracket U_1 \rrbracket \times \dots \times \llbracket U_m \rrbracket \rightarrow R^{\llbracket T \rrbracket}$$

Now, by currying we obtain:

$$\mathbf{\Lambda}(\widetilde{M}) : R^{\llbracket T_1 \rrbracket} \times \dots \times R^{\llbracket T_n \rrbracket} \rightarrow R^{\llbracket T \rrbracket \times \llbracket U_1 \rrbracket \times \dots \times \llbracket U_m \rrbracket}$$

and we have the following result:

$$\mathbf{\Lambda}(\widetilde{M}) = M$$

where the brackets on the left represent the interpretation of $\lambda^{R \times +}$ in the cartesian “ R -closed” category \mathcal{C} , and the brackets on the right represent the interpretation of $\lambda\mu$ -calculus in the category of continuations $R^{\mathcal{C}}$.

On the equational side, the equations of $\lambda^{R \times +}$ are given in Figure 10, where the two terms are of the same type: Since these equations are typed, M is of type 1 in (η_1^{λ}) . If M and N are $\lambda\mu$ -terms of the same type, then $M = N$ holds using the equations (β_{\rightarrow}) , (η_{\rightarrow}) , (ζ_{\rightarrow}) , (β_{\times}) , (η_{\times}) , (ζ_{\times}) , (β_0) , (η_0) and (ζ_0) of section 2.1.3 if and only if $\widetilde{M} = \widetilde{N}$ holds using the equations $(\beta_{\rightarrow}^{\lambda})$, $(\eta_{\rightarrow}^{\lambda})$, $(\beta_{\times}^{\lambda})$, $(\eta_{\times}^{\lambda})$, (β_{+}^{λ}) , (η_{+}^{λ}) and (η_1^{λ}) .

Just as we did for $\lambda\mu$ -calculus and categories of continuations, we use terms of $\lambda^{R \times +}$ with parameters in \mathcal{C} . For example, if $\mathfrak{K} : \prod_{j \in J} A_j \rightarrow B$ and $\mathfrak{U} : \prod_{j \in J} A_j \rightarrow R^B$, then $\mathfrak{U} \mathfrak{K} : \prod_{j \in J} A_j \rightarrow R$, where formally $\mathfrak{U} \mathfrak{K}$ is the term with parameters $(y x) \{ \mathfrak{K}/x, \mathfrak{U}/y \}$.

$$\begin{array}{l}
 [\langle \tilde{\phi}, \mathfrak{K} \rangle] \psi = [\mathfrak{K}] \psi \phi \quad \text{if} \left\{ \begin{array}{l} \phi : \prod_{j \in J} R^{A_j} \rightarrow R^B \mathfrak{Y} \left(\mathfrak{Y}_{k \in K} R^{D_k} \right) \\ \psi : \prod_{j \in J} R^{A_j} \rightarrow R^{R^B \times C} \mathfrak{Y} \left(\mathfrak{Y}_{k \in K} R^{D_k} \right) \\ \mathfrak{K} : \left(\prod_{j \in J} R^{A_j} \right) \times \left(\prod_{k \in K} D_k \right) \rightarrow C \end{array} \right. \\
 [*] \phi = \phi \quad \text{if} \phi : \prod_{j \in J} R^{A_j} \rightarrow R^1 \mathfrak{Y} \left(\mathfrak{Y}_{k \in K} R^{D_k} \right) \\
 [\text{in}_i \mathfrak{K}] \phi = [\mathfrak{K}] \pi_i \phi \quad \text{if} \left\{ \begin{array}{l} \phi : \prod_{j \in J} R^{A_j} \rightarrow (R^{B_1} \times R^{B_2}) \mathfrak{Y} \left(\mathfrak{Y}_{k \in K} R^{D_k} \right) \\ \mathfrak{K} : \left(\prod_{j \in J} R^{A_j} \right) \times \left(\prod_{k \in K} D_k \right) \rightarrow B_i \end{array} \right.
 \end{array}$$

 Figure 11: Interactions between \mathcal{C} and $R^{\mathcal{C}}$

2.2.4. *Interactions between \mathcal{C} and $R^{\mathcal{C}}$.* We will also extend the $\lambda\mu$ -terms with parameters of section 2.2.2 by allowing the substitution of terms of $\lambda^{R^{\times+}}$ with parameters in \mathcal{C} (defined in section 2.2.3) for μ -variables of $\lambda\mu$ -terms with parameters in $R^{\mathcal{C}}$. For example, if $\phi : \prod_{j \in J} R^{A_j} \rightarrow R^B \mathfrak{Y} (\mathfrak{Y}_{k \in K} R^{C_k})$ in $R^{\mathcal{C}}$ and $\mathfrak{K} : \left(\prod_{j \in J} R^{A_j} \right) \times \left(\prod_{k \in K} C_k \right) \rightarrow B$ in \mathcal{C} , then $[\mathfrak{K}] \phi : \prod_{j \in J} R^{A_j} \rightarrow R^1 \mathfrak{Y} (\mathfrak{Y}_{k \in K} R^{C_k})$ in $R^{\mathcal{C}}$. As expected, we can prove the equations of Figure 11. This possibility of having terms of $\lambda^{R^{\times+}}$ inside the brackets of $\lambda\mu$ -terms is closely related to the $\bar{\lambda}\mu\tilde{\mu}$ -calculus of [Her95, CH00], extended to handle products (using sums of $\lambda^{R^{\times+}}$).

3. A REALIZABILITY MODEL FOR CLASSICAL ANALYSIS

In this section, we define a realizability model which is based on the duality between realizers and counter-realizers. First, we explain the relativization needed to interpret the recursion scheme and the axiom of choice, then we define the realizability relation and we prove its adequacy for first-order logic, Peano arithmetic and the axiom of choice. Finally, we use our orthogonality-based model to extract computational content from classical proofs.

3.1. The relativization predicate. In the following we will consider two kinds of quantifications: the uniform and the relativized ones. From the point of view of provability in the two theories PA^ω and CA^ω considered in this article, this will not change anything. Indeed, we will define for PA^ω and CA^ω (which have uniform quantifications only) their relativized versions PA^{ω_r} and CA^{ω_r} (which have both uniform and relativized quantifications). Then we will describe a mapping from PA^ω (resp. CA^ω) to PA^{ω_r} (resp. CA^{ω_r}), and we will perform our realizability interpretation in the relativized theories.

Relativization is a technique that was already used in Krivine's models [Kri94, OS08, Miq11], and its utility will appear more clearly in the realizability interpretation. A realizer of $\forall x A$ (uniform quantification) is an element which must be a realizer of $A\{a/x\}$ for every instance a of x , whereas a realizer of $\forall^r x A$ (relativized quantification) is a function that turns any instance a of x into a realizer of $A\{a/x\}$. The uniform quantifiers suffice to realize the rules of first-order logic and Leibniz equality, but when it comes to Peano arithmetic, and more particularly to the axiom scheme of induction, the recursor of Gödel's system T needs to know on which particular natural number to recurse. A simple and intuitive example arises when we use the induction scheme to perform case-analysis: we prove on one hand $A\{0\}$, and on the other hand $A\{n\}$ for $n \neq 0$, so by case-analysis we get $\forall^r x A$ (note the relativized quantification). Then the corresponding program must read the natural number, test if it is zero, and then branch to the corresponding program. One solution would have been to relativize all quantifiers, however relativizing only when it is necessary gives a better understanding of the realizability interpretation and simpler extracted programs.

We explain now how we map proofs of a theory into proofs of a relativized version of the theory. Fix a theory $\mathcal{A}x$ on a signature Σ . Let Σ^r be the signature obtained by adding to Σ a predicate symbol for each base sort ι of Σ :

$$(\cdot^\iota)$$

We lift it to every sort with the following syntactic sugar:

$$(\iota^{\sigma \rightarrow \tau}) \triangleq \forall x^\sigma (\iota x^\sigma) \Rightarrow (\iota x^\tau) \quad (\text{so in particular } (\cdot^{\sigma \rightarrow \tau})^* = (\cdot^\sigma)^* \rightarrow (\cdot^\tau)^*)$$

First, since $\Sigma \subseteq \Sigma^r$, $\mathcal{A}x$ can be considered as a theory on Σ^r . We also define syntactic sugar for relativized quantifications:

$$\forall^r x^\sigma A \triangleq \forall x^\sigma (\iota x^\sigma \Rightarrow A) \quad \exists^r x^\sigma A \triangleq \neg \forall^r x^\sigma \neg A$$

and we define inductively the relativization A^r (on the signature Σ^r) of a formula A (on the signature Σ):

$$\begin{aligned} P(t_1^{\sigma_1}, \dots, t_n^{\sigma_n})^r &\triangleq P(t_1^{\sigma_1}, \dots, t_n^{\sigma_n}) & \perp^r &\triangleq \perp \\ (A \Rightarrow B)^r &\triangleq A^r \Rightarrow B^r & (A \wedge B)^r &\triangleq A^r \wedge B^r & (\forall x^\sigma A)^r &\triangleq \forall^r x^\sigma A^r \end{aligned}$$

This translation is extended to contexts the obvious way, so we write Γ^r and Δ^r .

Before going through the relativization of a theory, we first parameterize the syntax of our logic by splitting the set of basic predicates between positive and negative predicates. The negative predicates are those P for which $\neg \neg P \Rightarrow P$ is valid under the realizability interpretation, and this notion is fundamental when it comes to relativized theories. Indeed, while the inequality predicate of PA^ω is negative, the relativization predicate is positive and cannot be defined from a negative one.

Formally, we extend the definition of the signature of a logical system by distinguishing the negative predicate symbols (written P^-) from the positive ones (written P^+). We then extend this to all formulas of the logic by defining the following subsets of negative and positive formulas:

$$\begin{aligned} A^-, B^- &::= P^-(t_1^{\sigma_1}, \dots, t_n^{\sigma_n}) \mid \perp \mid A \Rightarrow B^- \mid A^- \wedge B^- \mid \forall x^\sigma A^- \\ A^+, B^+ &::= P^+(t_1^{\sigma_1}, \dots, t_n^{\sigma_n}) \mid A \Rightarrow B^+ \mid A^+ \wedge B^+ \mid A \wedge B^+ \mid \forall x^\sigma A^+ \end{aligned}$$

where the absence of polarity means that it can be either. In usual interpretations of classical proofs every formula has both a positive and a negative interpretation, which are orthogonal

to each other. This is reflected on the computational side by the duality between terms and contexts. The relativization predicate, however, only has a positive interpretation. In order for our logical rules to be valid under the realizability interpretation, we restrict our sequents by asking the right-hand context to contain only negative formulas. Formally, in a sequent:

$$\Gamma \vdash A \mid \Delta$$

the formulas appearing in Δ must all be negative, which will be emphasized from now on by writing Δ^- . There are no other requirements, so A and formulas of Γ can be either positive or negative. This restriction on the syntax of the sequents roughly amounts to forbid the structural rules on the right for positive formulas. In particular $\vdash \perp \Rightarrow A$ and $\vdash \neg\neg A \Rightarrow A$ are provable only if A is a negative formula. This restriction will appear as necessary in the proof of the adequacy lemma for first-order logic 3.10.

One could think at first sight that these polarities are related to the ones of LC [Gir91], however there is an important mismatch in the case of implication, where in LC $A \Rightarrow B$ is negative iff A is positive or B is positive, while in our setting it has the same polarity as B . After a discussion with Olivier Laurent, it appeared that the polarities defined here are more related to the system LU [Gir93], where our positive formulas are LU formulas with polarity 0, our negative formulas are LU formulas with polarity -1 , our \Rightarrow is LU's \supset , our \wedge is LU's $\&$ and our $\forall x$ is LU's $\bigwedge x$. A sequent $\Gamma \vdash A \mid \Delta$ in our setting then corresponds to a sequent $;\Gamma \vdash \Delta; A$ in LU.

For equational theories, the inequality predicate is defined to be negative. Since the inequality predicate is the only predicate of $\Sigma_{PA^\omega} = \Sigma_{CA^\omega}$, all the formulas written on this signature are negative, and therefore the proofs in PA^ω and CA^ω automatically respect the negativeness condition.

Since the negative predicates do not carry computational content, we also fix their interpretation to be the empty type of $\lambda\mu$ -calculus:

$$P^{-*} \triangleq 0$$

We now describe the process of relativization of a theory. In this process, the new predicate symbol $(\cdot)_!$ is defined to be positive, while the other predicates of Σ keep their polarity. First, an important fact is that the relativization of formulas preserves polarity:

Lemma 3.1. *If A^- (resp. A^+) is a negative (resp. positive) formula over the signature Σ , then $A^{-!}$ (resp. $A^{+!}$) is a negative (resp. positive) formula over the signature $\Sigma^!$ (obtained by adding a positive relativization predicate $(\cdot)_!$ for each base sort ι).*

Proof. By induction on A^- (resp. A^+). □

A relativized version of a theory $\mathcal{A}x$ on Σ is then a set of axioms $\mathcal{A}x^!$ on the signature $\Sigma^!$ such that if A is provable in $\mathcal{A}x$, then $A^!$ is provable in $\mathcal{A}x^!$. Be careful however that $\mathcal{A}x^!$ is in general different from $\{A^! \mid A \in \mathcal{A}x\}$. The following lemma gives a sufficient condition for $\mathcal{A}x^!$ to be a relativized version of $\mathcal{A}x$:

Lemma 3.2. *If $\mathcal{A}x^!$ is a theory on $\Sigma^!$ such that:*

- *for any constant individual c^σ of Σ , $\mathcal{A}x^! \vdash (c^\sigma)_!$*
- *for each $A \in \mathcal{A}x$, $\mathcal{A}x^! \vdash A^!$*
- *for each sort σ of Σ , there is a closed term t^σ*

then for any closed formula A on Σ :

$$\mathcal{A}x \vdash A \quad \Longrightarrow \quad \mathcal{A}x^! \vdash A^!$$

$$\begin{array}{c}
\frac{\Gamma \vdash A^- \mid A^-, \Delta^-}{\Gamma \vdash \perp \mid A^-, \Delta^-} \rightsquigarrow \frac{(\vec{x}^{\vec{\sigma}}), \Gamma^r \vdash A^{-r} \mid A^{-r}, \Delta^{-r}}{(\vec{x}^{\vec{\sigma}}), \Gamma^r \vdash \perp \mid A^{-r}, \Delta^{-r}} \\
\frac{\Gamma \vdash \perp \mid A^-, \Delta^-}{\Gamma \vdash A^- \mid \Delta^-} \rightsquigarrow \frac{(\vec{x}^{\vec{\sigma}}), \Gamma^r \vdash \perp \mid A^{-r}, \Delta^{-r}}{(\vec{x}^{\vec{\sigma}}), \Gamma^r \vdash A^{-r} \mid \Delta^{-r}} \\
\frac{\Gamma, A \vdash B \mid \Delta^-}{\Gamma \vdash A \Rightarrow B \mid \Delta^-} \rightsquigarrow \frac{(\vec{x}^{\vec{\sigma}}), \Gamma^r, A^r \vdash B^r \mid \Delta^{-r}}{(\vec{x}^{\vec{\sigma}}), \Gamma^r \vdash A^r \Rightarrow B^r \mid \Delta^{-r}} \\
\frac{\Gamma \vdash A \Rightarrow B \mid \Delta^- \quad \Gamma \vdash A \mid \Delta^-}{\Gamma \vdash B \mid \Delta^-} \rightsquigarrow \frac{(\vec{x}^{\vec{\sigma}}), \Gamma^r \vdash A^r \Rightarrow B^r \mid \Delta^{-r} \quad (\vec{x}^{\vec{\sigma}}), \Gamma^r \vdash A^r \mid \Delta^{-r}}{(\vec{x}^{\vec{\sigma}}), \Gamma^r \vdash B^r \mid \Delta^{-r}} \\
\frac{\Gamma \vdash A \mid \Delta^- \quad \Gamma \vdash B \mid \Delta^-}{\Gamma \vdash A \wedge B \mid \Delta^-} \rightsquigarrow \frac{(\vec{x}^{\vec{\sigma}}), \Gamma^r \vdash A^r \mid \Delta^{-r} \quad (\vec{x}^{\vec{\sigma}}), \Gamma^r \vdash B^r \mid \Delta^{-r}}{(\vec{x}^{\vec{\sigma}}), \Gamma^r \vdash A^r \wedge B^r \mid \Delta^{-r}} \\
\frac{\Gamma \vdash A_1 \wedge A_2 \mid \Delta^-}{\Gamma \vdash A_i \mid \Delta^-} \rightsquigarrow \frac{(\vec{x}^{\vec{\sigma}}), \Gamma^r \vdash A_1^r \wedge A_2^r \mid \Delta^{-r}}{(\vec{x}^{\vec{\sigma}}), \Gamma^r \vdash A_i^r \mid \Delta^{-r}}
\end{array}$$

Figure 12: Relativization of the rules for logical connectives

Proof. We translate inductively every proof in the theory $\mathcal{A}x$ (on the signature Σ) of a sequent:

$$\Gamma \vdash A \mid \Delta^-$$

to a proof in $\mathcal{A}x^r$ (on the signature Σ^r) of a sequent:

$$(\vec{x}^{\vec{\sigma}}), \Gamma^r \vdash A^r \mid \Delta^{-r}$$

where $\text{FV}(\Gamma, A, \Delta) \subseteq \vec{x}^{\vec{\sigma}}$. The translation of a proof $\vdash A$ in $\mathcal{A}x$ for A a closed formula is then a proof in $\mathcal{A}x^r$ of a sequent $(\vec{x}^{\vec{\sigma}}) \vdash A^r$. Indeed, some dummy variables may appear during the translation, without appearing free in A (this is due to our system not satisfying the subformula property). In order to eliminate these, we use the last hypothesis: from $(\vec{x}^{\vec{\sigma}}) \vdash A^r$ where A (and therefore A^r) is closed we can derive $\vdash \forall^r \vec{x}^{\vec{\sigma}} A^r$, and then $\vdash (\vec{t}^{\vec{\sigma}}) \Rightarrow A^r$ where $\vec{t}^{\vec{\sigma}}$ are closed individuals obtained from the last hypothesis. Finally, using lemma 3.3 below, we can combine the proof of $\vdash (\vec{t}^{\vec{\sigma}}) \Rightarrow A^r$ with proofs of $(\vec{t}^{\vec{\sigma}})$ and get a proof of $\vdash A^r$.

The translation of the axiom rule follows from the hypotheses of the lemma, the identity rule is translated as follows, with $\vec{x}^{\vec{\sigma}} = \text{FV}(\Gamma, A, \Delta)$:

$$\frac{}{\Gamma, A \vdash A \mid \Delta^-} \rightsquigarrow \frac{}{(\vec{x}^{\vec{\sigma}}), \Gamma^r, A^r \vdash A^r \mid \Delta^{-r}}$$

The rules for logical connectives are translated trivially as in Figure 12, where in the cases of introduction of implication and conjunction, the two premises can get the same relativized variables on the left by applying the (admissible) left weakening rule when necessary. The translation of the introduction of universal quantification is given by:

$$\frac{\Gamma \vdash A \mid \Delta^-}{\Gamma \vdash \forall y^r A \mid \Delta^-} (y^r \notin \text{FV}(\Gamma, \Delta^-)) \rightsquigarrow \frac{\frac{(y^r), (\vec{x}^{\vec{\sigma}}), \Gamma^r \vdash A^r \mid \Delta^{-r}}{(\vec{x}^{\vec{\sigma}}), \Gamma^r \vdash (y^r) \Rightarrow A^r \mid \Delta^{-r}}}{(\vec{x}^{\vec{\sigma}}), \Gamma^r \vdash \forall y^r ((y^r) \Rightarrow A^r) \mid \Delta^{-r}} (y^r \notin \text{FV}((\vec{x}^{\vec{\sigma}}), \Gamma^r, \Delta^{-r}))$$

and preserves the fact that all the free variables of a sequent are relativized in the context. Finally, in order to translate the elimination of the universal quantification we must prove that we can lift relativization to all constructs on individuals. This can be obtained using the hypothesis of the lemma requiring that for every individual constant c^σ of Σ , $\mathcal{A}x^r \sim \llbracket c^\sigma \rrbracket$. Indeed, we have the following lemma:

Lemma 3.3. *Suppose that for every individual constant c^σ of Σ , $\mathcal{A}x^r \sim \llbracket c^\sigma \rrbracket$. Let t^σ be an individual of the logic with free variables $\vec{x}^{\vec{\nu}}$. We have:*

$$\mathcal{A}x^r \sim \forall^r \vec{x}^{\vec{\nu}} \llbracket t^\sigma \rrbracket$$

Proof. We prove it by induction on t^σ . If t^σ is some c^σ , then this is an assumption of the lemma. If t^σ is a variable x^σ , then $\forall^r x^\sigma \llbracket x^\sigma \rrbracket \equiv \forall x^\sigma (\llbracket x^\sigma \rrbracket \Rightarrow \llbracket x^\sigma \rrbracket)$, which is trivially derivable, and if t^σ is $u^{\tau \rightarrow \sigma} v^\tau$, then we get $\llbracket \vec{x}^{\vec{\nu}} \rrbracket \vdash \forall y^\tau (\llbracket y^\tau \rrbracket \Rightarrow \llbracket (u y)^\sigma \rrbracket)$ and $\llbracket \vec{x}^{\vec{\nu}} \rrbracket \vdash \llbracket v^\tau \rrbracket$ from the induction hypotheses, so we obtain $\llbracket \vec{x}^{\vec{\nu}} \rrbracket \vdash \llbracket (u v)^\sigma \rrbracket$ and $\vdash \forall^r \vec{x}^{\vec{\nu}} \llbracket (u v)^\sigma \rrbracket$. \square

Now we can describe the translation of the elimination rule of the universal quantifier:

$$\frac{\Gamma \vdash \forall y^\tau A \mid \Delta^-}{\Gamma \vdash A \{t^\sigma/y^\tau\} \mid \Delta^-} \rightsquigarrow \frac{\frac{\llbracket \vec{x}^{\vec{\sigma}} \rrbracket, \Gamma^r \vdash \forall^r y^\tau A^r \mid \Delta^{-r}}{\llbracket \vec{x}^{\vec{\sigma}} \rrbracket, \Gamma^r \vdash \llbracket t^\sigma \rrbracket \Rightarrow A \{t/y\}^r \mid \Delta^{-r}} \quad \vdots}{\llbracket \vec{x}^{\vec{\sigma}} \rrbracket, \Gamma^r \vdash \llbracket t^\sigma \rrbracket \mid \Delta^{-r}}}{\llbracket \vec{x}^{\vec{\sigma}} \rrbracket, \Gamma^r \vdash A \{t/y\}^r \mid \Delta^{-r}}$$

where we can suppose without loss of generality that $\text{FV}(t^\sigma) \subseteq \vec{x}^{\vec{\sigma}}$ (using the admissible left weakening rule if necessary), and $\llbracket \vec{x}^{\vec{\sigma}} \rrbracket, \Gamma^r \vdash \llbracket t^\sigma \rrbracket \mid \Delta^r$ is easily derivable using $\forall^r \vec{z}^{\vec{\nu}} \llbracket t^\sigma \rrbracket$ (with $\vec{z}^{\vec{\nu}} = \text{FV}(t^\sigma) \subseteq \vec{x}^{\vec{\sigma}}$) from lemma 3.3.

Finally, the negativeness of the formulas in the right-hand context is preserved through the relativization, thanks to lemma 3.1. \square

3.1.1. *Relativized PA^ω : $PA^{\omega r}$.* We now define the relativized version $PA^{\omega r}$ of PA^ω , to which we apply lemma 3.2. First, $\Sigma_{PA^{\omega r}}$ is $\Sigma_{PA^\omega}^r$, that is Σ_{PA^ω} augmented with the positive predicate symbol $\llbracket t^\iota \rrbracket$. The axioms of $PA^{\omega r}$ are those of PA^ω (unrelativized) where the induction scheme is replaced with:

$$(\text{ind}^r) \quad \forall \vec{y} (A \{0/x\} \Rightarrow \forall^r x (A \Rightarrow A \{S x/x\}) \Rightarrow \forall^r x A)$$

plus the axioms $\llbracket 0^\iota \rrbracket$ and $\llbracket S^{\iota \rightarrow \iota} \rrbracket$. Notice that (ind^r) is different from $(\text{ind})^r$ (the parameters \vec{y} are not relativized). In order to use lemma 3.2 we need the following lemmas:

Lemma 3.4.

$$PA^{\omega r} \sim \llbracket \text{s} \rrbracket \quad PA^{\omega r} \sim \llbracket \text{k} \rrbracket \quad PA^{\omega r} \sim \llbracket \text{rec} \rrbracket$$

Proof. The formulas $\llbracket \text{s} \rrbracket$ and $\llbracket \text{k} \rrbracket$ are provable using the axioms of equality and respectively (Δs) and (Δk) . Indeed, $\llbracket \text{s} \rrbracket$ is the following formula:

$$\forall x^{\sigma \rightarrow \tau \rightarrow \nu} (\llbracket x \rrbracket \Rightarrow \forall y^{\sigma \rightarrow \tau} (\llbracket y \rrbracket \Rightarrow \forall z^\sigma (\llbracket z \rrbracket \Rightarrow \llbracket \text{s} x y z \rrbracket)))$$

which is equivalent in first-order logic to:

$$\forall x^{\sigma \rightarrow \tau \rightarrow \nu} \forall y^{\sigma \rightarrow \tau} \forall z^\sigma (\llbracket x \rrbracket \Rightarrow \llbracket y \rrbracket \Rightarrow \llbracket z \rrbracket \Rightarrow \llbracket \text{s} x y z \rrbracket)$$

using (Δs) and *(Leib)* this is equivalent to:

$$\forall x^{\sigma \rightarrow \tau \rightarrow \nu} \forall y^{\sigma \rightarrow \tau} \forall z^\sigma (\llbracket x \rrbracket \Rightarrow \llbracket y \rrbracket \Rightarrow \llbracket z \rrbracket \Rightarrow \llbracket x z (y z) \rrbracket)$$

and since $\langle x \rangle$ and $\langle y \rangle$ are the following formulas:

$$\forall x_1^\sigma (\langle x_1 \rangle \Rightarrow \forall x_2^\tau (\langle x_2 \rangle \Rightarrow \langle x x_1 x_2 \rangle)) \quad \forall y_1^\sigma (\langle y_1 \rangle \Rightarrow \langle y y_1 \rangle)$$

we can instantiate these with $x_1 \equiv z$, $x_2 \equiv y z$ and $y_1 \equiv z$ to obtain a proof of $\langle s \rangle$. The case of $\langle k \rangle$ is similar.

Similarly, $\langle \text{rec} \rangle$ is provable using the axioms of equality, $(\Delta \text{rec}0)$, $(\Delta \text{rec}S)$ and (ind^r) . $\langle \text{rec} \rangle$ is equivalent in first-order logic to:

$$\forall x^\sigma \forall y^{\iota \rightarrow \sigma \rightarrow \sigma} (\langle x \rangle \Rightarrow \forall^r y_1^\iota \forall y_2^\sigma (\langle y_2 \rangle \Rightarrow \langle y y_1 y_2 \rangle) \Rightarrow \forall^r z^\iota (\text{rec } x y z))$$

if we instantiate y_2 with $\text{rec } x y y_1$ it is sufficient to prove:

$$\forall x^\sigma \forall y^{\iota \rightarrow \sigma \rightarrow \sigma} (\langle x \rangle \Rightarrow \forall^r y_1^\iota (\langle \text{rec } x y y_1 \rangle \Rightarrow \langle y y_1 (\text{rec } x y y_1) \rangle) \Rightarrow \forall^r z^\iota (\text{rec } x y z))$$

which is equivalent, using $(\Delta \text{rec}0)$, $(\Delta \text{rec}S)$ and (Leib) , to:

$$\forall x^\sigma \forall y^{\iota \rightarrow \sigma \rightarrow \sigma} (\langle \text{rec } x y 0 \rangle \Rightarrow \forall^r y_1^\iota (\langle \text{rec } x y y_1 \rangle \Rightarrow \langle \text{rec } x y (S y_1) \rangle) \Rightarrow \forall^r z^\iota (\text{rec } x y z))$$

which is an instance of (ind^r) with the formula $\langle \text{rec } x y z \rangle$.

Moreover, these proofs of $\langle s \rangle$, $\langle k \rangle$ and $\langle \text{rec} \rangle$ respect the condition of having only negative formulas in the right-hand context, since this context is empty (which means that the proofs are valid in minimal logic). \square

Lemma 3.5. *For any $A \in PA^\omega$, $PA^{\omega^r} \vdash A^r$.*

Proof. For (Δs) , (Δk) , $(\Delta \text{rec}0)$, $(\Delta \text{rec}S)$ and (Snz) it follows from the fact that the formula $\forall x A \Rightarrow \forall^r x A$ is derivable in first-order logic. For (ind) it comes from this and the fact that the following formula:

$$\forall \vec{y} (A^r \{0/x\} \Rightarrow \forall^r x (A^r \Rightarrow A^r \{Sx/x\}) \Rightarrow \forall^r x A^r)$$

is an instance of (ind^r) . Here again, the right-hand context is empty so the proofs are correct. \square

Lemma 3.6. *For every sort σ on Σ_{PA^ω} , there is a closed individual t^σ .*

Proof. 0^σ from section 1.3 is such a term. \square

Using these three lemmas, it follows from lemma 3.2 that for any closed formula A on the signature Σ_{PA^ω} :

$$PA^\omega \vdash A \quad \Longrightarrow \quad PA^{\omega^r} \vdash A^r$$

3.1.2. Interpreting PA^{ω^r} in system $T + \Omega$. System T was introduced by Gödel in [Göd58] in order to give a consistency proof of Heyting arithmetic (and therefore of Peano arithmetic by double-negation translation). This system can be equivalently formulated as a system of primitive recursive functionals, which is an extension of primitive recursive functions to higher types. It is strictly more powerful than primitive recursion, since for example the Ackermann's function is expressible in system T.

System T has one base type I for natural numbers, product and function types, constants for 0 and successor, and a recursion operator of type $T \rightarrow (I \rightarrow T \rightarrow T) \rightarrow I \rightarrow T$ for any type T . Restricting the type of the recursor to $T = I$ gives back the usual primitive recursive functions.

$$\begin{array}{ll}
 M_{(\mathit{refl})} \triangleq \lambda x.x : 0 \rightarrow 0 & M_{(\Delta \mathit{k})} \triangleq \lambda x.x : 0 \rightarrow 0 \\
 M_{(\mathit{Leib})} \triangleq \lambda x.x : (A^* \rightarrow 0) \rightarrow A^* \rightarrow 0 & M_{(\Delta \mathit{s})} \triangleq \lambda x.x : 0 \rightarrow 0 \\
 M_{(\mathit{Snz})} \triangleq \Omega : 0 & M_{(\Delta \mathit{rec0})} \triangleq \lambda x.x : 0 \rightarrow 0 \\
 M_{(\mathit{indr})} \triangleq \mathit{rec} : B^* \rightarrow (I \rightarrow B^* \rightarrow B^*) \rightarrow I \rightarrow B^* & M_{(\Delta \mathit{recS})} \triangleq \lambda x.x : 0 \rightarrow 0 \\
 M_{(\mathit{k})} \triangleq \lambda xy.x : T \rightarrow U \rightarrow T & M_{(\mathit{s})} \triangleq \lambda xyz.x z (y z) : (T \rightarrow U \rightarrow V) \rightarrow (T \rightarrow U) \rightarrow T \rightarrow V \\
 M_{(\mathit{0})} \triangleq \bar{0} : I & M_{(\mathit{S})} \triangleq \mathit{succ} : I \rightarrow I \quad M_{(\mathit{rec})} \triangleq \mathit{rec} : T \rightarrow (I \rightarrow T \rightarrow T) \rightarrow I \rightarrow T
 \end{array}$$

 Figure 13: Interpretation of the axioms of $PA^{\omega r}$

Since μ PCF contains constants for every natural number, successor, predecessor and general recursion, it is easy to encode system T in it. Indeed, if we define:

$$\mathit{rec} \triangleq \lambda xy.Y (\lambda zu.\mathit{if}_0 u x (y (\mathit{pred} u)) (z (\mathit{pred} u)))$$

Then it is easy to derive:

$$\mathit{rec} : T \rightarrow (I \rightarrow T \rightarrow T) \rightarrow I \rightarrow T$$

and in order to prove that it implements Gödel's recursor we must prove that it satisfies the corresponding equations:

Lemma 3.7. *Let $M : T$ and $N : I \rightarrow T \rightarrow T$. We have:*

$$\mathit{rec} M N \bar{0} = M$$

and for any $n \in \mathbb{N}$:

$$\mathit{rec} M N \overline{n+1} = N \bar{n} (\mathit{rec} M N \bar{n})$$

Proof. This follows easily from the definition of rec and the equations of μ PCF for pred , if_0 and Y . \square

Therefore in the following we will consider system T as a subsystem of μ PCF. We will also use the constant Ω to interpret the fact that 0 is not a successor. In order to interpret $PA^{\omega r}$ in system T + Ω , we first fix the interpretation of the relativization predicate:

$$(\cdot^{\iota})^* \triangleq I$$

The inequality predicate being a negative one, it is interpreted as 0. Finally, we provide a term M_A of type A^* for each $A \in PA^{\omega r}$ in Figure 13, where B is a formula with free variables among $x^{\iota}, \vec{y}^{\sigma}$.

3.1.3. Relativized axiom of choice. As we did for PA^{ω} , we define the relativized version $CA^{\omega r}$ of CA^{ω} . First, the signature is the same as $\Sigma_{PA^{\omega r}}$: it is $\Sigma_{CA^{\omega}} = \Sigma_{PA^{\omega}}$ augmented with a positive predicate symbol (\cdot^{ι}) . In this section, A denotes a formula over $\Sigma_{CA^{\omega r}} = \Sigma_{PA^{\omega r}}$ with free variables among $x^{\iota}, y^{\sigma}, z^{\sigma}, \vec{u}^{\vec{\tau}}$. For clarity, we write $A \{t, u, v\}$ instead of $A \{t/x, u/y, v/z\}$. The axioms of $CA^{\omega r}$ are those of $PA^{\omega r}$ plus the following version of dependent choice:

$$(\mathit{DC}^r) \forall \vec{u}^{\vec{\tau}} (\forall^r x^{\iota} \forall^r y^{\sigma} (\forall z^{\sigma} \neg A \{x, y, z\} \Rightarrow \forall x'^{\iota} A \{x', y, y\}) \Rightarrow \exists v^{\iota \rightarrow \sigma} \forall^r x^{\iota} A \{x, v x, v (\mathit{S} x)\})$$

where $A\{x, y, z\}$ is of the shape $(|z|) \wedge C$. The formula A in (DC^r) could be understood as an abbreviation, and the axiom is schematic in C . This version is quite different from $(DC)^r$ which is:

$$\forall^r \vec{u}^r (\forall^r x^t \forall^r y^\sigma \exists^r z^\sigma B^r \{x, y, z\} \Rightarrow \exists^r v^{t \rightarrow \sigma} \forall^r x^t B^r \{x, v x, v(Sx)\})$$

where B is a formula over $\Sigma_{CA^\omega} = \Sigma_{PA^\omega}$ with free variables among $x^t, y^\sigma, z^\sigma, \vec{u}^r$, for which we use again the notation $B^r \{t, u, v\}$. First, some quantifications have been unrelativized, but another important difference is that we replaced \perp with $\forall x' A\{x', y, y\}$. This change is in the spirit of [EO12] and will allow an easier realizability interpretation in section 3.3.3. In order to use lemma 3.2, we need to prove that $(DC)^r$ is derivable in $CA^{\omega r}$. We will actually prove that the instance of (DC^r) with:

$$A\{x, y, z\} \equiv (|z|) \wedge (|y|) \wedge B^r \{x, y, z\}$$

implies $(DC)^r$ in first-order logic. This is indeed an instance of (DC^r) since A is of the shape $(|z|) \wedge C$. Since $\forall \vec{u} C \Rightarrow \forall^r \vec{u} C$ is derivable, it is sufficient to prove the following lemma:

Lemma 3.8. *The following formula over the signature $\Sigma_{PA^{\omega r}} = \Sigma_{CA^{\omega r}}$:*

$$\begin{aligned} \forall^r x \forall^r y (\forall z \neg A\{x, y, z\} \Rightarrow \forall x' A\{x', y, y\}) &\Rightarrow \forall v \neg \forall^r x A\{x, v x, v(Sx)\} \Rightarrow \perp \\ &\Rightarrow \\ \forall^r x \forall^r y \exists^r z B^r \{x, y, z\} &\Rightarrow \forall^r v \neg \forall^r x B^r \{x, v x, v(Sx)\} \Rightarrow \perp \end{aligned}$$

where $A\{x, y, z\} \equiv (|z|) \wedge (|y|) \wedge B^r \{x, y, z\}$, is provable in the logical system with relativization defined in section 3.1.

Proof. We do this by proving the following two sequents:

$$\begin{aligned} \forall^r x \forall^r y \exists^r z B^r \{x, y, z\} \vdash \forall^r x \forall^r y (\forall z \neg A\{x, y, z\} \Rightarrow \forall x' A\{x', y, y\}) \\ \forall^r v \neg \forall^r x B^r \{x, v x, v(Sx)\} \vdash \forall v \neg \forall^r x A\{x, v x, v(Sx)\} \end{aligned}$$

- For the first one, we suppose $\forall^r x \forall^r y \exists^r z B^r \{x, y, z\}$, $(|x|)$, $(|y|)$ and $\forall z \neg A\{x, y, z\}$, and we want to prove $(|y|) \wedge (|y|) \wedge B^r \{x', y, y\}$. $(|y|)$ is an hypothesis and we deduce $B^r \{x', y, y\}$ from \perp (which is valid since $B^r \{x', y, y\}$ is a negative formula by lemma 3.1 and therefore $\perp \Rightarrow B^r \{x', y, y\}$ is derivable) by applying the hypothesis $\forall^r x \forall^r y (\forall z \neg B^r \{x, y, z\} \Rightarrow \perp)$ with $(|x|)$ and $(|y|)$, so the only thing left to prove is $\forall^r z \neg B^r \{x, y, z\}$. We derive it from $\forall z \neg A\{x, y, z\}$ and $(|y|)$, which amounts to proving the following sequent:

$$\forall z ((|z|) \wedge (|y|) \wedge B^r \{x, y, z\} \Rightarrow \perp), (|y|) \vdash \forall z ((|z|) \Rightarrow B^r \{x, y, z\} \Rightarrow \perp)$$

which is immediate

- The second sequent can be rewritten as:

$$\forall v ((|v|) \Rightarrow \forall^r x B^r \{x, v x, v(Sx)\} \Rightarrow \perp) \vdash \forall v (\forall^r x A\{x, v x, v(Sx)\} \Rightarrow \perp)$$

Therefore, it is sufficient to prove:

$$\forall^r x A\{x, v x, v(Sx)\} \vdash (|v|) \wedge \forall^r x B^r \{x, v x, v(Sx)\}$$

which is after unfolding some definitions:

$$\forall^r x ((|v(Sx)|) \wedge (|v x|) \wedge B^r \{x, v x, v(Sx)\}) \vdash \forall^r x (|v x|) \wedge \forall^r x B^r \{x, v x, v(Sx)\}$$

and this sequent is indeed provable in our logical system. \square

Thanks to this lemma, we can now apply lemma 3.2 to get for any A on the signature Σ_{CA^ω} :

$$CA^\omega \vdash A \quad \Longrightarrow \quad CA^{\omega^r} \vdash A^r$$

3.1.4. Interpreting the axiom of choice with bar recursion. Bar recursion is an operator which can be seen as recursion on well-founded trees. It was first introduced by Spector in [Spe62] to extend Gödel's Dialectica interpretation to Heyting arithmetic augmented with the axiom of countable choice. This operator was studied in [Koh90], and a more uniform operator which is very similar to bar recursion was introduced in [BBC98] and used in a realizability setting. A version in which the well-foundedness of trees is implicit was proposed in [BO05] under the name of modified bar recursion, and we use this version here. For a comparison between these different forms of bar recursion and other similar principles we refer the reader to [Pow13, Pow14].

In this section, we first encode lists and list operators in μ PCF. Then we define the modified bar recursion operator of [BO05] that we will use to provide our computational interpretation of the axiom of dependent choice.

Encoding lists in μ PCF. In order to define bar recursion, we first need to encode lists and operations on lists in μ PCF. We choose to represent a list by a natural number (the size of the list) together with a (partial) function on natural numbers. We define the type of lists and notations for the size of a list, the empty list, and the access to a particular element:

$$T^\diamond \triangleq I \times (I \rightarrow T) \quad |M| \triangleq \pi_1 M \quad \epsilon \triangleq \langle \bar{0}, \lambda x. \Omega \rangle \quad M \downarrow N \triangleq \pi_2 M N$$

If $M : T^\diamond$ and $N : I$, then it is easy to prove:

$$|M| : I \quad \epsilon : T^\diamond \quad |\epsilon| = \bar{0} \quad M \downarrow N : T \quad \epsilon \downarrow N = \Omega$$

In order to define extensions of lists, we need subtraction on natural numbers $\mathbf{sub} : I \rightarrow I \rightarrow I$ and tests of equality $\mathbf{if}_= : I \rightarrow I \rightarrow T \rightarrow T \rightarrow T$ and strict ordering $\mathbf{if}_< : I \rightarrow I \rightarrow T \rightarrow T \rightarrow T$, which we can define in μ PCF. These operators satisfy the following equations for $m, n \in \mathbb{N}$:

$$\mathbf{sub} \bar{m} \bar{n} = \begin{cases} \overline{m - n} & \text{if } n \leq m \\ \bar{0} & \text{otherwise} \end{cases} \quad \mathbf{if}_= \bar{m} \bar{n} M N = \begin{cases} M & \text{if } m = n \\ N & \text{otherwise} \end{cases} \quad \mathbf{if}_< \bar{m} \bar{n} M N = \begin{cases} M & \text{if } m < n \\ N & \text{otherwise} \end{cases}$$

We are now able to define the extension of a list by a single element:

$$M * N \triangleq \langle \mathbf{succ} |M|, \lambda x. \mathbf{if}_= x |M| N (M \downarrow x) \rangle$$

as expected, if $M : T^\diamond$ and $N : T$ then $M * N : T^\diamond$, $|M * N| = \mathbf{succ} |M|$ and $(M * N) \downarrow |M| = N$. Finally, we define infinite extension of a list with a constant element:

$$M @ N \triangleq \lambda x. \mathbf{if}_< x |M| (M \downarrow x) N$$

We can derive from $M : T^\diamond$ and $N : T$ that $M @ N : I \rightarrow T$ and:

$$(M_0 * M_1 * \dots * M_{n-1} @ N) \downarrow \bar{m} = \begin{cases} M_m & \text{if } m < n \\ N & \text{otherwise} \end{cases}$$

The bar recursion operator. We have now all necessary material to define formally the bar recursion operator:

$$\mathbf{barrec} \triangleq \lambda uv. \mathbf{Y} (\lambda zy. v (y @ (u y (\lambda x. z (y * x))))))$$

Bar recursor can be typed as expected:

$$\mathbf{barrec} : (T^\diamond \rightarrow (T \rightarrow U) \rightarrow T) \rightarrow ((I \rightarrow T) \rightarrow U) \rightarrow T^\diamond \rightarrow U$$

And it verifies indeed the equation:

$$\mathbf{barrec} M N P = N (P @ M P (\lambda x. \mathbf{barrec} M N (P * x)))$$

Interpreting dependent choice using bar recursion. We use here the bar-recursion operator to provide the term $M_{(\mathbf{DC}^r)}$ interpreting the axiom of dependent choice. Remember that (\mathbf{DC}^r) is:

$$\forall \vec{u}^{\vec{r}} (\forall^r x^\iota \forall^r y^\sigma (\forall z^\sigma \neg A \{x, y, z\} \Rightarrow \forall x'^\iota A \{x', y, y\}) \Rightarrow \exists v^{\iota \rightarrow \sigma} \forall^r x^\iota A \{x, v x, v (\mathbf{S} x)\})$$

where A is a formula over $\Sigma_{CA^{\omega^r}} = \Sigma_{PA^{\omega^r}}$ with free variables among $x^\iota, y^\sigma, z^\sigma, \vec{u}^{\vec{r}}$ and which is of the shape $(z) \wedge C$. As in the previous section, we write $A \{t, u, v\}$ instead of $A \{t/x, u/y, v/z\}$. The type of $M_{(\mathbf{DC}^r)}$, $(\mathbf{DC}^r)^*$ is:

$$M_{(\mathbf{DC}^r)} : (I \rightarrow T \rightarrow (A^* \rightarrow 0) \rightarrow A^*) \rightarrow ((I \rightarrow A^*) \rightarrow 0) \rightarrow 0$$

In order to define $M_{(\mathbf{DC}^r)}$ we make an informal reasoning. Suppose $M : \iota \rightarrow T \rightarrow (A^* \rightarrow 0) \rightarrow A^*$ is a witness of:

$$\forall^r x^\iota \forall^r y^\sigma (\forall z^\sigma \neg A \{x, y, z\} \Rightarrow \forall x'^\iota A \{x', y, y\})$$

and $N : (I \rightarrow A^*) \rightarrow 0$ is a witness of:

$$\forall v^{\iota \rightarrow \sigma} \neg \forall^r x^\iota A \{x, v x, v (\mathbf{S} x)\}$$

We want to build from this, using \mathbf{barrec} , a witness of \perp . We will use the following instance of \mathbf{barrec} :

$$\mathbf{barrec} : (A^{*\diamond} \rightarrow (A^* \rightarrow 0) \rightarrow A^*) \rightarrow ((I \rightarrow A^*) \rightarrow 0) \rightarrow A^{*\diamond} \rightarrow 0$$

The idea now is that \mathbf{barrec} will build a sequence of witnesses of $A \{x, v x, v (\mathbf{S} x)\}$. The first argument represents the recursive step. If we have an element $w : A^\diamond$ which represents the sequence of witnesses already computed, then M computes the next element of w , given its length and last element. Here we have two cases, the first one is when $|w| = \bar{0}$, so we must initialize the sequence with an element of type $(\cdot^\sigma)^*$ that we can choose arbitrarily. Since σ is a sort of the logic, we can write it as $\sigma_1 \rightarrow \dots \rightarrow \sigma_n \rightarrow \iota$ so:

$$(\cdot^\sigma)^* = (\cdot^{\sigma_1})^* \rightarrow \dots \rightarrow (\cdot^{\sigma_n})^* \rightarrow I$$

and we define this arbitrary element to be $\lambda x_1 \dots x_n. \bar{0}$. In the following, we will simply write this term as $\bar{0}$, leaving the type implicit. In the second case, the last element of w is $w \downarrow \mathbf{pred} |w| \downarrow$. Therefore, we provide M with $|w|$ and $\mathbf{if}_0 |w| \bar{0} (\pi_1 w \downarrow \mathbf{pred} |w| \downarrow)$. The π_1 is because the last element of w is a witness of $A \{n-2, v(n-2), v(n-1)\}$ where n is the length of w , so since A is of shape $(z) \wedge C$, $\pi_1 w \downarrow \mathbf{pred} |w| \downarrow$ is a witness of $(v(n-1))$. The first argument is then:

$$\lambda w. M |w| (\mathbf{if}_0 |w| \bar{0} (\pi_1 w \downarrow \mathbf{pred} |w| \downarrow))$$

The second argument represents the behavior if we have an infinite sequence of witnesses $w : I \rightarrow B$. In that case we simply provide N with this argument w , so the third argument is just N . Finally, the last argument of `barrec` is the initial sequence of witnesses, that is the empty sequence ϵ . We have then:

$$\text{barrec} (\lambda w.M \mid w \mid (\text{if}_0 \mid w \mid \bar{0} (\pi_1 w \mid \text{pred} \mid w \mid \downarrow))) N \epsilon : 0$$

The interpretation of (DC^r) is therefore defined as:

$$M_{(DC^r)} \triangleq \lambda x y. \text{barrec} (\lambda v.x \mid v \mid (\text{if}_0 \mid v \mid \bar{0} (\pi_1 v \mid \text{pred} \mid v \mid \downarrow))) y \epsilon$$

3.2. The realizability relation.

3.2.1. Negative translation and orthogonality. The first realizability models for classical logic were obtained by combining Gödel's negative translation with intuitionistic realizability [BBC98, BO05, Koh08]. Gödel's negative translation from Peano arithmetic PA_{ω}^{ω} (equivalent to PA^{ω}) to HA_{ω}^{ω} (see section 1.3.1) maps a formula A to A^{\neg} by prefixing inductively all the positive connectives and atomic predicates of A with a double negation. It holds that if $PA_{\omega}^{\omega} \Vdash A$, then $HA_{\omega}^{\omega} \Vdash A^{\neg}$, and the proof of this relies on the fact that for every axiom A of PA_{ω}^{ω} , $HA_{\omega}^{\omega} \Vdash A \Rightarrow A^{\neg}$. Therefore, a realizability model for PA_{ω}^{ω} can be obtained from a realizability model for HA_{ω}^{ω} using Gödel's negative translation. Concerning the extraction of witnesses, if $PA_{\omega}^{\omega} \Vdash \exists x^t (t = 0)$, then $HA_{\omega}^{\omega} \Vdash \neg\neg\exists x^t \neg\neg (t = 0)$ from which we easily get $HA_{\omega}^{\omega} \Vdash \neg\neg\exists x^t (t = 0)$. While in usual intuitionistic realizability the formula \perp has no realizer so the model is sound, Friedman's trick is to allow \perp to have realizers. If we then take the realizers of \perp to be the same as those of $\exists x^t (t = 0)$, then combining the proof of $\neg\neg\exists x^t (t = 0)$ with the identity gives a realizer of $\exists x^t (t = 0)$, and therefore the witness.

In Krivine's [Kri09] classical realizability models this double step (negative translation + intuitionistic realizability) is avoided through the use of orthogonality in system F. In these models there is a set of terms Λ and a set of stacks Π . Each formula has a set of realizers (the truth value, subset of Λ) and a set of counter-realizers (the falsity value, subset of Π). The falsity values are primitive, an orthogonality relation is defined between Λ and Π , and the truth values are defined as the orthogonals of the falsity values, so they are orthogonally closed.

Here we work in a typed setting so we must choose the types of realizers and counter-realizers so they can interact. Given a formula A , the set of realizers of A would normally be a set of morphisms in a category of continuations R^C from the terminal object $\mathbf{1}$ to the interpretation of A : $\llbracket A^* \rrbracket = R^{\llbracket A^* \rrbracket}$, and under the duality between terms and contexts of $\lambda\mu$ -calculus, a natural choice for the counter-realizers of A is a set of morphisms in C from $\mathbf{1}$ to $\llbracket A^* \rrbracket$. Then we can combine a potential realizer of A with a potential counter-realizer using the evaluation morphism $\mathbf{ev} : R^{\llbracket A^* \rrbracket} \times \llbracket A^* \rrbracket \rightarrow R$ so we obtain a morphism from $\mathbf{1}$ to R . The potential realizer and counter-realizer are orthogonal to each other or not, depending on the result. Therefore, in order to define a non-trivial orthogonality relation, there must be at least two morphisms from $\mathbf{1}$ to R , and if we want to perform extraction this homset has to be isomorphic to the set of values we want to extract. In realizability for arithmetic, this is usually done by choosing an object R which is the same as the interpretation of natural numbers, however this choice has some drawbacks. Indeed, some computational models can be naturally seen as categories of continuations for a given R , and this R may not be

isomorphic to the object of natural numbers. Take for example the model of Hyland-Ong games [HO00]. We know that by relaxing the well-bracketing condition we obtain a fully abstract model of μ PCF [Lai97]. It is therefore not a surprise that the same game model is a category of continuations R^C , and it turns out that in this category of continuations, the object R is the one-move arena. The key point for this is that the arena of natural numbers is the exponential of the one-move arena by the countable product of one-move arenas, details can be found in [Blo14], section 4.4. But since there is only one strategy on the one-move arena (the empty strategy), we cannot easily define a non-trivial orthogonality relation.

Therefore we choose here a different approach and rely on Friedman's trick directly in the definition of the realizability relation: our orthogonality relation relies on an artificially added output channel. Formally we add a μ -variable in the process of interpreting logic in $\lambda\mu$ -calculus. A proof:

$$\frac{\mathfrak{p}}{A_1, \dots, A_n \vdash A \mid B_1^-, \dots, B_m^-}$$

is now translated to a $\lambda\mu$ -term:

$$x_1 : A_1^*, \dots, x_n : A_n^* \vdash \mathfrak{p}^* : A^* \mid \alpha_1 : B_1^*, \dots, \alpha_n : B_m^*, \kappa : Z$$

where Z is some fixed base type, by applying the (admissible) rule of right weakening of $\lambda\mu$ -calculus. The μ -variable κ is, intuitively, a continuation variable which can be used by a realizer or a counter-realizer to stop computation and give an answer. Apart from its use in the definition of the realizability relation, this feature will also be used in the proof of the extraction result, for which Z will be instantiated with the type of natural numbers. After translating the proof \mathfrak{p} to a $\lambda\mu$ -term \mathfrak{p}^* , we interpret it in R^C as a morphism:

$$\mathfrak{p}^* \in R^C([A_1^*] \times \dots \times [A_n^*], [A^*] \wp [B_1^*] \wp \dots \wp [B_m^*] \wp [Z])$$

In the particular case of empty contexts, \mathfrak{p}^* is a morphism in $R^C(\mathbf{1}, [A^*] \wp [Z])$, and we therefore choose the potential realizers of a closed formula A to be such morphisms. Similarly we choose the potential counter-realizers of A to be morphisms in $\mathcal{C}(\llbracket Z \rrbracket, \llbracket A^* \rrbracket)$. As explained in section 2.2.2 and by taking the convention that the potential realizers have a free μ -variable κ of type Z , we use the syntax of $\lambda\mu$ -calculus (and possibly $[\kappa]$ and $\mu\kappa$) to manipulate these. We also substitute morphisms of \mathcal{C} for μ -variables, as in section 2.2.4, so if $\phi \in R^C(\mathbf{1}, [A^*] \wp [Z])$ is a potential realizer, and if $\mathfrak{k} \in \mathcal{C}(\llbracket Z \rrbracket, \llbracket A^* \rrbracket)$ is a potential counter-realizer, then $[\mathfrak{k}] \phi \in R^C(\mathbf{1}, [0] \wp [Z])$, since $[A^*] = R^{\llbracket A^* \rrbracket}$ and $[0] = R$, and therefore $\mu\kappa. [\mathfrak{k}] \phi \in R^C(\mathbf{1}, [Z])$. Since the object $[Z] = R^{\llbracket Z \rrbracket}$ can (and will) be chosen larger than $[0] = R$, we can now define when \mathfrak{k} is orthogonal to ϕ depending on $\mu\kappa. [\mathfrak{k}] \phi$. Typically, in the model of unbracketed games, R is the one-move arena and $\llbracket Z \rrbracket$ is the countable product of the one-move arena, so $[Z] = R^{\llbracket Z \rrbracket}$ is indeed the usual arena of natural numbers.

The choice of having a separate μ -variable instead of choosing a big enough object R can also provide a simpler interpretation of proofs in R^C . We give a comparison of our interpretation of the identity proof of $\perp \Rightarrow \perp$ in arithmetic with the usual interpretation in figure 14, where we take R^C to be the unbracketed games model. We can see that with the usual interpretation where R is the object of natural numbers, this proof is interpreted as a strategy from natural numbers to natural numbers. However, in our interpretation the same proof is interpreted as a strategy on the arena $[0]^{\llbracket 0 \rrbracket} \wp [Z]$ where $[0]$ is the one-move arena and $[Z]$ is the arena of natural numbers (indeed, the \wp operation on arenas is the merge of roots). This simplification doesn't happen in every model however. For example, in Scott

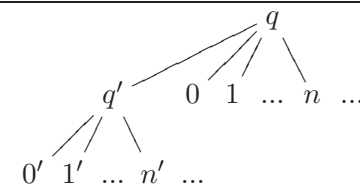
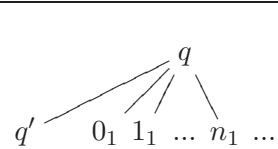
	usual interpretation:	our interpretation:
arena:		
views of the strategy:	$\{\epsilon; qq'\} \cup \{qq'mm' \mid m \in \mathbb{N}\}$	$\{\epsilon; qq'\}$

Figure 14: Comparison of our interpretation with the usual one

domains, the natural choice of taking the singleton set for R leads to a degenerated model, since R^A is then isomorphic to R for any domain A . In order to get a non-degenerated model, we would need to choose for R a bigger domain, and therefore we would lose the benefit of our simpler interpretation.

In [BR13] the goals of orthogonally-defined realizability and extraction were achieved similarly by taking realizers of a formula A to be interpretations of closed terms of type $\llbracket Z \rrbracket \rightarrow [A^*]$, but also interpreting the atomic formulas and \perp with the base type Z instead of the empty type 0 , which led to unnecessary complex types for the realizers.

To summarize, in order to have the benefit of the simpler interpretation, our categorical model R^C needs to have an object R which is significantly simpler than $[Z] = R^{\llbracket Z \rrbracket}$, and $[Z]$ should be large enough to interpret natural numbers. This rules out Scott domains, but unbracketed games models do have this property. It would be interesting to find other examples of such models, the main candidates being Laird's bistable biorders [Lai07] and Berry and Curien's sequential algorithms [BC82], which are both fully abstract models of μ PCF.

3.2.2. Truth values, falsity values. We fix a first-order signature Σ and a Σ -structure \mathcal{M} . We also fix a corresponding $\lambda\mu$ signature and a type P^* in this signature for each positive predicate P of Σ , as in section 2.1.2. We interpret $\lambda\mu$ -calculus in a category of continuations R^C as in section 2.2.2 and we use the syntax of $\lambda\mu$ -calculus to describe strategies of R^C , so we omit the interpretation brackets. All the $\lambda\mu$ -terms that we write from now on are to be understood as morphisms in R^C .

In order to build our realizability relation by orthogonality, and later on to perform extraction on Π_2^0 formulas using Friedman's trick, our model is parameterized with a set

$$\perp \subseteq R^C(\mathbf{1}, [Z])$$

which is, intuitively, the set of “correct” values that can be output through the variable κ .

We now define the set of realizers of a formula, that we call its truth value. We fix for each positive predicate P of Σ and each $a_1, \dots, a_n \in \sigma_1^{\mathcal{M}} \times \dots \times \sigma_n^{\mathcal{M}}$ a set of morphisms $|P(a_1, \dots, a_n)| \subseteq R^C(\mathbf{1}, [P^*] \wp [Z])$. The extension of this to every closed formula A on Σ with parameters in \mathcal{M} is given in Figure 15, so $|A| \subseteq R^C(\mathbf{1}, [A^*] \wp [Z])$. Remark that contrary to [Kri09], we do not define the truth values as the orthogonals of the falsity values. In our logical system, only some predicates are negative, and therefore only some formulas are negative. In the realizability interpretation, only the negative formulas are given a falsity

$$\begin{aligned}
|A \Rightarrow B| &= \{\phi \mid \forall \psi \in |A|, \phi \psi \in |B|\} & |\forall x^\sigma A| &= \bigcap_{a \in \sigma^{\mathcal{M}}} |A \{a/x\}| \\
|A \wedge B| &= \{\phi \mid \pi_1 \phi \in |A| \wedge \pi_2 \phi \in |B|\} & |\perp| &= \{\phi \mid \mu\kappa.\phi \in \perp\} \\
|P^-(a_1, \dots, a_n)| &= \begin{cases} R^{\mathcal{C}}(\mathbf{1}, [0] \wp [Z]) & \text{if } \mathcal{M} \models P^-(a_1, \dots, a_n) \\ \{\phi \mid \mu\kappa.\phi \in \perp\} & \text{otherwise} \end{cases}
\end{aligned}$$

Figure 15: Truth values

$$\begin{aligned}
\|A \Rightarrow B^-\| &= \{\langle \tilde{\phi}, \mathfrak{K} \rangle \mid \phi \in |A| \wedge \mathfrak{K} \in \|B^-\|\} & \|\forall x^\sigma A^-\| &= \bigcup_{a \in \sigma^{\mathcal{M}}} \|A^-\{a/x\}\| \\
\|A^- \wedge B^-\| &= \{\text{in}_1 \mathfrak{K} \mid \mathfrak{K} \in \|A^-\|\} \cup \{\text{in}_2 \mathfrak{K} \mid \mathfrak{K} \in \|B^-\|\} & \|\perp\| &= \{*\} \\
\|P^-(a_1, \dots, a_n)\| &= \begin{cases} \emptyset & \text{if } \mathcal{M} \models P^-(a_1, \dots, a_n) \\ \{*\} & \text{otherwise} \end{cases}
\end{aligned}$$

Figure 16: Falsity values

value and their truth values will be proved to be orthogonal to their falsity values. For the other formulas the truth value is primitive and may not be bi-orthogonally closed.

For every closed negative formula A^- with parameters in \mathcal{M} the falsity value of A^- is given in Figure 16, so $\|A^-\| \subseteq \mathcal{C}(\llbracket Z \rrbracket, \llbracket A^* \rrbracket)$. As explained in section 2.2.3, we use here the syntax of $\lambda^{R \times +}$ to manipulate morphisms in \mathcal{C} . We define now an orthogonality relation between $R^{\mathcal{C}}(\mathbf{1}, [A^*] \wp [Z])$ and $\mathcal{C}(\llbracket Z \rrbracket, \llbracket A^* \rrbracket)$: if $\phi \in R^{\mathcal{C}}(\mathbf{1}, [A^*] \wp [Z])$ and $\mathfrak{K} \in \mathcal{C}(\llbracket Z \rrbracket, \llbracket A^* \rrbracket)$, then $\mu\kappa.[\mathfrak{K}] \phi \in R^{\mathcal{C}}(\mathbf{1}, [Z])$, so we define:

$$\phi \perp \mathfrak{K} \triangleq \mu\kappa.[\mathfrak{K}] \phi \in \perp$$

The following lemma states that the truth values are indeed the orthogonals of the falsity values defined above for negative formulas:

Lemma 3.9. *For every closed negative formula A^- with parameters in \mathcal{M} :*

$$|A^-| = \{\phi \mid \forall \mathfrak{K} \in \|A^-\|, \phi \perp \mathfrak{K}\}$$

Proof. We prove this result by induction on the structure of the formula:

- \perp : if $\phi \in R^{\mathcal{C}}(\mathbf{1}, [\perp^*] \wp [Z])$ then:

$$\phi \in |\perp| \Leftrightarrow \mu\kappa.\phi \in \perp \Leftrightarrow \mu\kappa.[*] \phi \in \perp \Leftrightarrow \phi \perp *$$

from which we conclude since $\|\perp\| = \{*\}$.

- $P^-(a_1, \dots, a_n)$: if $\mathcal{M} \models P^-(a_1, \dots, a_n)$ then the result is immediate, and otherwise the proof is the same as for \perp .
- $A^- \wedge B^-$: if $\phi \in R^{\mathcal{C}}(\mathbf{1}, ([A^*] \times [B^*]) \wp [Z])$, $\mathfrak{K}_1 \in \mathcal{C}(\llbracket Z \rrbracket, \llbracket A^* \rrbracket)$, $\mathfrak{K}_2 \in \mathcal{C}(\llbracket Z \rrbracket, \llbracket B^* \rrbracket)$, then $\mu\kappa.[\text{in}_i \mathfrak{K}_i] \phi = \mu\kappa.[\mathfrak{K}_i] \pi_i \phi$, therefore:

$$\pi_i \phi \perp \mathfrak{K}_i \Leftrightarrow \mu\kappa.[\mathfrak{K}_i] \pi_i \phi \in \perp \Leftrightarrow \mu\kappa.[\text{in}_i \mathfrak{K}_i] \phi \in \perp \Leftrightarrow \phi \perp \text{in}_i \mathfrak{K}_i$$

and finally:

$$\begin{aligned}
 \sigma \in |A^- \wedge B^-| &\Leftrightarrow \begin{cases} \pi_1 \phi \in |A^-| \\ \pi_2 \phi \in |B^-| \end{cases} \\
 &\Leftrightarrow \begin{cases} \forall \mathfrak{K}_1 \in \|A^-\|, \pi_1 \phi \perp \mathfrak{K}_1 \\ \forall \mathfrak{K}_2 \in \|B^-\|, \pi_2 \phi \perp \mathfrak{K}_2 \end{cases} \text{ by induction hypothesis} \\
 &\Leftrightarrow \begin{cases} \forall \mathfrak{K}_1 \in \|A^-\|, \phi \perp \text{in}_1 \mathfrak{K}_1 \\ \forall \mathfrak{K}_2 \in \|B^-\|, \phi \perp \text{in}_2 \mathfrak{K}_2 \end{cases} \\
 &\Leftrightarrow \forall \mathfrak{K} \in \|A^- \wedge B^-\|, \phi \perp \mathfrak{K}
 \end{aligned}$$

- $A \Rightarrow B^-$: if $\phi \in R^{\mathcal{C}}(\mathbf{1}, [B^*]^{[A^*]} \wp [Z])$, $\psi \in R^{\mathcal{C}}(\mathbf{1}, [A^*] \wp [Z])$ and $\mathfrak{K} \in \mathcal{C}(\llbracket Z \rrbracket, \llbracket B^* \rrbracket)$, then $\mu\kappa. [\langle \tilde{\psi}, \mathfrak{K} \rangle] \phi = [\mathfrak{K}] \phi \psi$, therefore:

$$\phi \psi \perp \mathfrak{K} \Leftrightarrow \mu\kappa. [\mathfrak{K}] \phi \psi \in \perp \Leftrightarrow \mu\kappa. [\langle \tilde{\psi}, \mathfrak{K} \rangle] \phi \in \perp \Leftrightarrow \phi \perp \langle \tilde{\psi}, \mathfrak{K} \rangle$$

and finally:

$$\begin{aligned}
 \phi \in |A \Rightarrow B^-| &\Leftrightarrow \forall \psi \in |A|, \phi \psi \in |B^-| \\
 &\Leftrightarrow \forall \psi \in |A|, \forall \mathfrak{K} \in \|B^-\|, \phi \psi \perp \mathfrak{K} \quad \text{by induction hypothesis} \\
 &\Leftrightarrow \forall \psi \in |A|, \forall a \in \|B^-\|, \phi \perp \langle \tilde{\psi}, a \rangle \\
 &\Leftrightarrow \forall \mathfrak{K}' \in \|A \Rightarrow B^-\|, \psi \perp \mathfrak{K}'
 \end{aligned}$$

- $\forall x^\sigma A^-$: if $\phi \in R^{\mathcal{C}}(\mathbf{1}, [A^*] \wp [Z])$, then:

$$\begin{aligned}
 \phi \in |\forall x^\sigma A^-| &\Leftrightarrow \forall a \in \sigma^{\mathcal{M}}, \phi \in |A^- \{a/x\}| \\
 &\Leftrightarrow \forall a \in \sigma^{\mathcal{M}}, \forall \mathfrak{K} \in \|A^- \{a/x\}\|, \phi \perp \mathfrak{K} \\
 &\Leftrightarrow \forall \mathfrak{K} \in \mathcal{C}(\llbracket Z \rrbracket, \llbracket A^* \rrbracket), (\exists a \in \sigma^{\mathcal{M}}, \mathfrak{K} \in \|A^- \{a/x\}\|) \Rightarrow \phi \perp \mathfrak{K} \\
 &\Leftrightarrow \forall \mathfrak{K} \in \|\forall x^\sigma A^-\|, \phi \perp \mathfrak{K} \quad \square
 \end{aligned}$$

3.3. Adequacy.

3.3.1. *Adequacy for first-order logic.* We will now state the adequacy lemma, which states the soundness of our realizability interpretation with respect to first-order classical logic. It is interesting to remark that the only cases which depends on the orthogonality relation are those of introduction and elimination of the \perp formula. It is not much of a surprise, since these rules are the ones that make our proof system classical. Most of the other cases are straightforward, though some care must be taken for the \forall rules.

In order to prove the adequacy lemma, we suppose that the interpretations of the terms associated to the axioms are realizers of these axioms: for every $A \in \mathcal{A}x$, $M_A \in |A|$. The adequacy lemma is then as follows:

Lemma 3.10. *Suppose \mathfrak{p} is a proof of $\Gamma \vdash A \mid \Delta^-$ with $FV(\Gamma, A, \Delta) \subseteq \bar{y}^{\vec{\sigma}}$, so:*

$$\vec{x} : \Gamma^* \vdash \mathfrak{p}^* : A^* \mid \vec{\alpha} : \Delta^*, \kappa : Z$$

then for any $\vec{a} \in \vec{\sigma}^{\mathcal{M}}, \vec{\phi} \in |\Gamma \{\vec{a}/\vec{y}\}|, \vec{\kappa} \in \|\Delta^- \{\vec{a}/\vec{y}\}\|$, we have:

$$\mathfrak{p}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} \in |A \{\vec{a}/\vec{y}\}|$$

In particular if A is a closed formula, $\mathfrak{p}^* \in |A|$

Proof. By induction on the proof tree:

- If \mathfrak{p} is the identity rule then \mathfrak{p}^* is:

$$\frac{}{\vec{x} : \Gamma^*, z : A^* \vdash z : A^* \mid \vec{\alpha} : \Delta^*, \kappa : Z}$$

Let $\vec{a} \in \vec{\sigma}^{\mathcal{M}}, \vec{\phi} \in |\Gamma \{\vec{a}/\vec{y}\}|, \psi \in |A \{\vec{a}/\vec{y}\}|$ and $\vec{\kappa} \in \|\Delta^- \{\vec{a}/\vec{y}\}\|$. Then we have:

$$\mathfrak{p}^* \left\{ \vec{\phi}/\vec{x}, \psi/z, \vec{\kappa}/\vec{\alpha} \right\} = \psi \in |A \{\vec{a}/\vec{y}\}|$$

- If \mathfrak{p} is the axiom rule then \mathfrak{p}^* is:

$$\vec{x} : \Gamma^* \vdash M_A : A^* \mid \vec{\alpha} : \Delta^*, \kappa : Z$$

Let $\vec{a} \in \vec{\sigma}^{\mathcal{M}}, \vec{\phi} \in |\Gamma \{\vec{a}/\vec{y}\}|$ and $\vec{\kappa} \in \|\Delta^- \{\vec{a}/\vec{y}\}\|$. Then we have:

$$\mathfrak{p}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} = M_A \in |A| = |A \{\vec{a}/\vec{y}\}|$$

since A is closed and $M_A \in |A|$ by assumption.

- If \mathfrak{p} ends with an introduction of \Rightarrow then \mathfrak{p}^* is:

$$\frac{\vec{x} : \Gamma^*, x : A^* \vdash \mathfrak{q}^* : B^* \mid \vec{\alpha} : \Delta^*, \kappa : Z}{\vec{x} : \Gamma^* \vdash \lambda x. \mathfrak{q}^* : A^* \rightarrow B^* \mid \vec{\alpha} : \Delta^*, \kappa : Z}$$

Let $\vec{a} \in \vec{\sigma}^{\mathcal{M}}, \vec{\phi} \in |\Gamma \{\vec{a}/\vec{y}\}|$ and $\vec{\kappa} \in \|\Delta^- \{\vec{a}/\vec{y}\}\|$. Then for any $\psi \in |A \{\vec{a}/\vec{y}\}|$ we have:

$$\left(\mathfrak{p}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} \right) \psi = \mathfrak{q}^* \left\{ \vec{\phi}/\vec{x}, \psi/y, \vec{\kappa}/\vec{\alpha} \right\} \in |B \{\vec{a}/\vec{y}\}|$$

by induction hypothesis, since $\psi \in |A \{\vec{a}/\vec{y}\}|$. Therefore:

$$\mathfrak{p}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} \in |(A \Rightarrow B) \{\vec{a}/\vec{y}\}|$$

- If \mathfrak{p} ends with an elimination of \Rightarrow then \mathfrak{p}^* is:

$$\frac{\vec{x} : \Gamma^* \vdash \mathfrak{q}^* : A^* \rightarrow B^* \mid \vec{\alpha} : \Delta^*, \kappa : Z \quad \vec{x} : \Gamma^* \vdash \mathfrak{r}^* : A^* \mid \vec{\alpha} : \Delta^*, \kappa : Z}{\vec{x} : \Gamma^* \vdash \mathfrak{q}^* \mathfrak{r}^* : B^* \mid \vec{\alpha} : \Delta^*, \kappa : Z}$$

Let $\vec{a} \in \vec{\sigma}^{\mathcal{M}}, \vec{\phi} \in |\Gamma \{\vec{a}/\vec{y}\}|$ and $\vec{\kappa} \in \|\Delta^- \{\vec{a}/\vec{y}\}\|$. Then we have:

$$\mathfrak{p}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} = \left(\mathfrak{q}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} \right) \left(\mathfrak{r}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} \right) \in |B \{\vec{a}/\vec{y}\}|$$

since by induction hypothesis:

$$\mathfrak{q}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} \in |(A \Rightarrow B) \{\vec{a}/\vec{y}\}| \quad \text{and} \quad \mathfrak{r}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} \in |A \{\vec{a}/\vec{y}\}|$$

- If \mathfrak{p} ends with an introduction of \wedge then \mathfrak{p}^* is:

$$\frac{\vec{x} : \Gamma^* \vdash \mathfrak{q}^* : A^* \mid \vec{\alpha} : \Delta^*, \kappa : Z \quad \vec{x} : \Gamma^* \vdash \mathfrak{r}^* : B^* \mid \vec{\alpha} : \Delta^*, \kappa : Z}{\vec{x} : \Gamma^* \vdash \langle \mathfrak{q}^*, \mathfrak{r}^* \rangle : A^* \times B^* \mid \vec{\alpha} : \Delta^*, \kappa : Z}$$

Let $\vec{a} \in \vec{\sigma}^{\mathcal{M}}, \vec{\phi} \in |\Gamma \{\vec{a}/\vec{y}\}|$ and $\vec{\kappa} \in \|\Delta^- \{\vec{a}/\vec{y}\}\|$. Then we have:

$$\mathfrak{p}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} = \left\langle \mathfrak{q}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\}, \mathfrak{r}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} \right\rangle \in |(A \wedge B) \{\vec{a}/\vec{y}\}|$$

since by induction hypothesis:

$$\mathfrak{q}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} \in |A \{\vec{a}/\vec{y}\}| \quad \text{and} \quad \mathfrak{r}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} \in |B \{\vec{a}/\vec{y}\}|$$

- If \mathfrak{p} ends with an elimination of \wedge then \mathfrak{p}^* is:

$$\frac{\vec{x} : \Gamma^* \vdash \mathfrak{q}^* : A_1^* \times A_2^* \mid \vec{\alpha} : \Delta^*, \kappa : Z}{\vec{x} : \Gamma^* \vdash \pi_i \mathfrak{q}^* : A_i^* \mid \vec{\alpha} : \Delta^*, \kappa : Z}$$

Let $\vec{a} \in \vec{\sigma}^{\mathcal{M}}$, $\vec{\phi} \in |\Gamma \{\vec{a}/\vec{y}\}|$ and $\vec{\kappa} \in \|\Delta^- \{\vec{a}/\vec{y}\}\|$. Then we have:

$$\mathfrak{p}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} = \pi_i \left(\mathfrak{q}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} \right) \in |A_i \{\vec{a}/\vec{y}\}|$$

since by induction hypothesis:

$$\mathfrak{q}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} \in |(A_1 \wedge A_2) \{\vec{a}/\vec{y}\}|$$

- If \mathfrak{p} ends with an introduction of \forall then \mathfrak{p}^* is:

$$\vec{x} : \Gamma^* \vdash \mathfrak{q}^* : (\forall z^{\tau} A)^* \mid \vec{\alpha} : \Delta^*, \kappa : Z \quad = \quad \vec{x} : \Gamma^* \vdash \mathfrak{q}^* : A^* \mid \vec{\alpha} : \Delta^*, \kappa : Z$$

Let $\vec{a} \in \vec{\sigma}^{\mathcal{M}}$, $\vec{\phi} \in |\Gamma \{\vec{a}/\vec{y}\}|$ and $\vec{\kappa} \in \|\Delta^- \{\vec{a}/\vec{y}\}\|$. Then for any $b \in \tau^{\mathcal{M}}$, $\vec{\phi} \in |\Gamma \{\vec{a}/\vec{y}, b/z\}|$ and $\vec{\kappa} \in \|\Delta^- \{\vec{a}/\vec{y}, b/z\}\|$ (since $z^{\tau} \notin \text{FV}(\Gamma, \Delta)$), so by induction hypothesis:

$$\mathfrak{q}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} \in |A \{\vec{a}/\vec{y}, b/z\}|$$

Therefore $\mathfrak{p}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} \in |(\forall z^{\tau} A) \{\vec{a}/\vec{y}\}|$.

- If \mathfrak{p} ends with an elimination of \forall then \mathfrak{p}^* is:

$$\vec{x} : \Gamma^* \vdash \mathfrak{q}^* : (A \{t^{\tau}/z^{\tau}\})^* \mid \vec{\alpha} : \Delta^*, \kappa : Z \quad = \quad \vec{x} : \Gamma^* \vdash \mathfrak{q}^* : (\forall z^{\tau} A)^* \mid \vec{\alpha} : \Delta^*, \kappa : Z$$

Let $\vec{a} \in \vec{\sigma}^{\mathcal{M}}$, $\vec{\phi} \in |\Gamma \{\vec{a}/\vec{y}\}|$ and $\vec{\kappa} \in \|\Delta^- \{\vec{a}/\vec{y}\}\|$. By induction hypothesis:

$$\mathfrak{q}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} \in \bigcap_{b \in \tau^{\mathcal{M}}} |A \{\vec{a}/\vec{y}, b/z\}|$$

so taking $b = (t \{\vec{a}/\vec{y}\})^{\mathcal{M}} \in \tau^{\mathcal{M}}$ we get:

$$\mathfrak{q}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} \in \left| A \left\{ \vec{a}/\vec{y}, (t \{\vec{a}/\vec{y}\})^{\mathcal{M}}/z \right\} \right|$$

and since $\mathfrak{p}^* = \mathfrak{q}^*$ and $A \left\{ \vec{a}/\vec{y}, (t \{\vec{a}/\vec{y}\})^{\mathcal{M}}/z \right\} = A \{t/z\} \{\vec{a}/\vec{y}\}$ we get:

$$\mathfrak{p}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} \in |A \{t/z\} \{\vec{a}/\vec{y}\}|$$

- If \mathfrak{p} ends with an introduction of \perp then \mathfrak{p}^* is:

$$\frac{\vec{x} : \Gamma^* \vdash \mathfrak{q}^* : A^* \mid \beta : A^*, \vec{\alpha} : \Delta^*, \kappa : Z}{\vec{x} : \Gamma^* \vdash [\beta] \mathfrak{q}^* : 0 \mid \beta : A^*, \vec{\alpha} : \Delta^*, \kappa : Z}$$

Let $\vec{a} \in \vec{\sigma}^{\mathcal{M}}$, $\vec{\phi} \in |\Gamma \{\vec{a}/\vec{y}\}|$, $\vec{\kappa} \in \|\Delta^- \{\vec{a}/\vec{y}\}\|$ and $\mathfrak{u} \in \|A^- \{\vec{a}/\vec{y}\}\|$. Then we have:

$$\begin{aligned} \mathfrak{p}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha}, \mathfrak{u}/\beta \right\} &= ([\beta] \mathfrak{q}^*) \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha}, \mathfrak{u}/\beta \right\} \\ &= [\mathfrak{u}] \left(\mathfrak{q}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha}, \mathfrak{u}/\beta \right\} \right) \in |\perp| = |\perp \{\vec{a}/\vec{y}\}| \end{aligned}$$

since by induction hypothesis:

$$\mathfrak{q}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha}, \mathfrak{u}/\beta \right\} \in |A^- \{ \vec{a}/\vec{y} \}|$$

so $\mathfrak{q}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha}, \mathfrak{u}/\beta \right\} \perp \mathfrak{u}$ by lemma 3.9, and $\mu\kappa. [\mathfrak{u}] \left(\mathfrak{q}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha}, \mathfrak{u}/\beta \right\} \right) \in \perp$.

- If \mathfrak{p} ends with an elimination of \perp then \mathfrak{p}^* is:

$$\frac{\vec{x} : \Gamma^* \vdash \mathfrak{q}^* : 0 \mid \beta : A^*, \vec{\alpha} : \Delta^*, \kappa : Z}{\vec{x} : \Gamma^* \vdash \mu\beta.\mathfrak{q}^* : A^* \mid \vec{\alpha} : \Delta^*, \kappa : Z}$$

Let $\vec{a} \in \vec{\sigma}^{\mathcal{M}}$, $\vec{\phi} \in |\Gamma \{ \vec{a}/\vec{y} \}|$ and $\vec{\kappa} \in \|\Delta^- \{ \vec{a}/\vec{y} \}\|$. Then for any $\mathfrak{u} \in \|A^- \{ \vec{a}/\vec{y} \}\|$ we have:

$$\begin{aligned} \mu\kappa. [\mathfrak{u}] \left(\mathfrak{p}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} \right) &= \mu\kappa. \left([\mathfrak{u}] \mu\beta.\mathfrak{q}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} \right) \\ &= \mu\kappa. \left(\mathfrak{q}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha}, \mathfrak{u}/\beta \right\} \right) \in \perp \end{aligned}$$

since by induction hypothesis, $\mathfrak{q}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha}, \mathfrak{u}/\beta \right\} \in |\perp \{ \vec{a}/\vec{y} \}| = |\perp|$. Therefore:

$$\mathfrak{p}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} \perp \mathfrak{u}$$

and so by lemma 3.9:

$$\mathfrak{p}^* \left\{ \vec{\phi}/\vec{x}, \vec{\kappa}/\vec{\alpha} \right\} \in |A^- \{ \vec{a}/\vec{y} \}| \quad \square$$

3.3.2. Adequacy for Peano arithmetic. We fix now the theory to be PA^{ω^r} on $\Sigma_{PA^{\omega^r}}$ (the inequality predicate being negative, and the relativization predicate being positive), and the structure \mathcal{M} to be a model of PA^ω , the inequality predicate being interpreted in \mathcal{M} as in section 1.5. For simplicity we write n for $(\mathbb{S}^n 0)^{\mathcal{M}}$ where $n \in \mathbb{N}$. The $\lambda\mu$ signature is that of μ PCF (see section 2.1.4) and we interpret the two predicates as in sections 3.1 and 3.1.2:

$$\neq_\sigma^* \triangleq 0 \quad (.)^* \triangleq I$$

and the axioms as in section 3.1.2. We suppose that the category of continuations R^C is a model of μ PCF (see definition 2.8). Since we want to extract algorithms on natural numbers, we fix $Z = I$.

The realizability value for the relativization predicate is:

$$|(a : \iota)| \triangleq \begin{cases} \{ \bar{n} \} & \text{if } a = n \text{ for some } n \in \mathbb{N} \\ \emptyset & \text{otherwise} \end{cases}$$

First, all equalities which are true in the model are trivially realized:

Lemma 3.11. *Let t^σ and u^σ be first-order terms with $FV(t, u) = \vec{x}^{\vec{\tau}}$.*

$$\text{If } \mathcal{M} \models \forall \vec{x}^{\vec{\tau}} (t = u) \text{ then } \lambda x.x \in \left| \forall \vec{x}^{\vec{\tau}} (t = u) \right|$$

Proof. Let $\vec{a} \in \vec{\tau}^{\mathcal{M}}$. Since $\mathcal{M} \models \forall \vec{x}^{\vec{\tau}} (t = u)$, we have $(t \{ \vec{a}/\vec{x} \})^{\mathcal{M}} = (u \{ \vec{a}/\vec{x} \})^{\mathcal{M}}$, so:

$$|t \{ \vec{a}/\vec{x} \} \neq u \{ \vec{a}/\vec{x} \}| = |\perp|$$

Therefore, for any $\phi \in |t \{ \vec{a}/\vec{x} \} \neq u \{ \vec{a}/\vec{x} \}|$:

$$(\lambda x.x) \phi = \phi \in |t \{ \vec{a}/\vec{x} \} \neq u \{ \vec{a}/\vec{x} \}| = |\perp|$$

and so:

$$\lambda x.x \in \left| \forall \vec{x}^{\vec{\tau}} (t \neq u \Rightarrow \perp) \right| = \left| \forall \vec{x}^{\vec{\tau}} (t = u) \right| \quad \square$$

Therefore, since \mathcal{M} is a model of PA^ω we have immediately the following results:

$$\begin{aligned} M_{(\mathbf{refl})} &= \lambda x.x \in |\forall x^\sigma (x =_\sigma x)| \\ M_{(\Delta s)} &= \lambda x.x \in |\forall x^{\sigma \rightarrow \tau \rightarrow \nu} \forall y^{\sigma \rightarrow \tau} \forall z^\sigma (s x y z =_\nu x z (y z))| \\ M_{(\Delta k)} &= \lambda x.x \in |\forall x^\sigma \forall y^\tau (k x y =_\sigma x)| \\ M_{(\Delta \mathbf{rec}0)} &= \lambda x.x \in |\forall x^\sigma \forall y^{\iota \rightarrow \sigma \rightarrow \sigma} (\mathbf{rec} x y 0 =_\sigma x)| \\ M_{(\Delta \mathbf{rec}S)} &= \lambda x.x \in |\forall x^\sigma \forall y^{\iota \rightarrow \sigma \rightarrow \sigma} \forall z^\iota (\mathbf{rec} x y (S z) =_\sigma y z (\mathbf{rec} x y z))| \end{aligned}$$

The non-confusion axiom and Leibniz scheme are easy:

Lemma 3.12.

$$\begin{aligned} M_{(\mathbf{S}nz)} &= \Omega \in |\forall x^\iota (S x \neq_\iota 0)| \\ M_{(\mathbf{Leib})} &= \lambda x.x \in \left| \forall \vec{z}^{\vec{\tau}} \forall x^\sigma \forall y^\sigma (\neg A \Rightarrow A \{y/x\} \Rightarrow x \neq_\sigma y) \right| \end{aligned}$$

Proof. Since $|\langle a \rangle| = \emptyset$ if $a \notin \mathbb{N}$, it is sufficient to prove that for any $n \in \mathbb{N}$, $\Omega \in |n + 1 \neq 0|$, which is true since $\mathcal{M} \models (\mathbf{S}nz)$ so $|n + 1 \neq 0| = R^C(\mathbf{1}, [0] \wp [I])$. Let now $\vec{a} \in \vec{\tau}^{\mathcal{M}}$, $b, c \in \sigma^{\mathcal{M}}$, $\phi \in |\neg A \{b/x, \vec{a}/\vec{z}\}|$ and $\psi \in |A \{c/x, \vec{a}/\vec{z}\}|$. If $b \neq c$, then $|b \neq c| = R^C(\mathbf{1}, [0] \wp [I])$ and therefore $(\lambda x.x) \phi \psi \in |b \neq c|$. Otherwise, $b = c$ so $|A \{b/x, \vec{a}/\vec{z}\}| = |A \{c/x, \vec{a}/\vec{z}\}|$ and by definition of $|A \Rightarrow B|$ we get $\phi \psi \in |\perp|$. Moreover, since $b = c$, $|b \neq c| = |\perp|$ and $\phi \psi \in |b \neq c|$. Finally we get $M_{(\mathbf{Leib})} \in |\forall \vec{z}^{\vec{\tau}} \forall x^\sigma \forall y^\sigma (\neg A \Rightarrow A \{y/x\} \Rightarrow x \neq_\sigma y)|$. \square

The adequacy for the induction axiom scheme is as follows:

Lemma 3.13.

$$M_{(\mathbf{ind}^r)} = \mathbf{rec} \in \left| \forall \vec{y}^{\vec{\tau}} (A \{0/x\} \Rightarrow \forall^r x^\iota (A \Rightarrow A \{S x/x\}) \Rightarrow \forall^r x^\iota A) \right|$$

Proof. Let $\vec{a} \in \vec{\tau}^{\mathcal{M}}$, $\phi \in |A \{0/x, \vec{a}/\vec{y}\}|$, $\psi \in |\forall^r x^\iota (A \{\vec{a}/\vec{y}\} \Rightarrow A \{S x/x, \vec{a}/\vec{y}\})|$. Since $|\langle b \rangle| = \emptyset$ if $b \notin \mathbb{N}$, we can use induction to prove that for any $n \in \mathbb{N}$, $\mathbf{rec} \phi \psi \bar{n} \in |A \{n/x, \vec{a}/\vec{y}\}|$:

- $n = 0$: $\mathbf{rec} \phi \psi \bar{0} = \phi \in |A \{0/x, \vec{a}/\vec{y}\}|$
- $n + 1$: $\mathbf{rec} \phi \psi \overline{n+1} = \mathbf{rec} \phi \psi \mathbf{succ} \bar{n} = \psi \bar{n} (\mathbf{rec} \phi \psi \bar{n})$. Since $\bar{n} \in |\langle n \rangle|$, we get:

$$\psi \bar{n} \in |A \{n/x, \vec{a}/\vec{y}\} \Rightarrow A \{n+1/x, \vec{a}/\vec{y}\}|$$

and since by induction hypothesis we have $\mathbf{rec} \phi \psi \bar{n} \in |A \{n/x, \vec{a}/\vec{y}\}|$ we get:

$$\psi \bar{n} (\mathbf{rec} \phi \psi \bar{n}) \in |A \{n+1/x, \vec{a}/\vec{y}\}| \quad \square$$

The last axioms are the relativization ones:

Lemma 3.14.

$$M_{(\mathbf{0})} = \bar{0} \in |\langle 0 \rangle| \quad M_{(\mathbf{S})} = \mathbf{succ} \in |\langle S \rangle|$$

Proof. The first one is immediate, since $|\langle 0 \rangle| = \{\bar{0}\}$. For the second one, remember that:

$$\langle S \rangle \equiv \forall^r x^\iota (\langle S x \rangle) \equiv \forall x^\iota (\langle x \rangle \Rightarrow \langle S x \rangle)$$

Since $|\langle a \rangle| = \emptyset$ if $a \notin \mathbb{N}$, it is sufficient to prove that for any $n \in \mathbb{N}$, $\mathbf{succ} \in |\langle n \rangle \Rightarrow \langle n+1 \rangle|$. Since $|\langle n \rangle| = \{\bar{n}\}$, it follows from $\mathbf{succ} \bar{n} = \overline{n+1} \in |\langle n+1 \rangle|$. \square

3.3.3. *Adequacy for the axiom of choice.* When it comes to the axiom of countable choice AC, the usual route of negative translation followed by intuitionistic realizability becomes much more difficult. Indeed, $AC \Rightarrow AC^\neg$ is not provable in intuitionistic logic, therefore the path described in section 3.2.1 cannot be followed as-is and an intuitionistic realizer of AC^\neg must be provided. In [BBC98], a variant of bar recursion was used to realize AC^\neg , while in [BO05] the principle of double negation shift $\forall x \neg\neg A \Rightarrow \neg\neg\forall x A$ was realized using bar recursion (see 3.1.4). With this principle, it becomes possible to derive $AC \Rightarrow AC^\neg$ in intuitionistic logic, and since AC is realized in intuitionistic models by the identity, one obtains a realizer of AC^\neg .

We follow here a different approach, since our realizability model is for classical logic, and we prove that bar recursion realizes the axiom of dependent choice (DC^r) in our classical model. The negative translation of proofs corresponds to the continuation-passing-style translation on terms and the semantics of $\lambda\mu$ -calculus in a category of continuations R^C corresponds to the semantics of its CPS-translation into $\lambda^{R\times+}$ in the cartesian “ R -closed” category \mathcal{C} as stated in section 2.2.3, so in order to compare more closely our model with the usual indirect realizability interpretation, one would need to define a realizability relation for intuitionistic logic directly in \mathcal{C} . Our choice of working in the category of continuations R^C is mainly motivated by the existence of computational models with a natural structure of category of continuations (as the unbracketed Hyland-Ong games, see section 3.2.1).

We suppose now that \mathcal{M} satisfies (DC), and we use the variant of the bar recursion operator defined in section 3.1.4 to realize the axiom of dependent choice (DC^r). First, we need to make some assumptions on our category of continuations. The first one is a continuity requirement:

Definition 3.15 (Continuity). If $\phi \in R^C(\mathbf{1}, [(I \rightarrow T) \rightarrow 0] \wp [I])$, $\psi \in R^C(\mathbf{1}, [I \rightarrow T] \wp [I])$, then there exists $m \in \mathbb{N}$ such that for any $\zeta \in R^C(\mathbf{1}, [I \rightarrow T] \wp [I])$:

$$(\forall m' < m, \zeta \overline{m'} = \psi \overline{m'}) \Rightarrow \mu\kappa.\phi\zeta = \mu\kappa.\phi\psi$$

This requirement is satisfied in particular by games models since in that case R^C is a cpo-enriched category in which the base types I and 0 are interpreted as flat domains. Note however that the full set-theoretic model doesn’t satisfy this, since we can for example consider a function which gives 0 if the input sequence is the constant 0 sequence, and 1 otherwise.

The second requirement states that we can construct a function from any sequence of elements:

Definition 3.16 (Sequence internalization). If $(\phi_n)_{n \in \mathbb{N}}$ is a sequence of morphisms in the homset $R^C(\mathbf{1}, [T] \wp [I])$, then there exists a morphism $\phi \in R^C(\mathbf{1}, [I \rightarrow T] \wp [I])$ such that for any $n \in \mathbb{N}$, $\phi \overline{n} = \phi_n$.

In particular, all functions on natural numbers must exist in the model, even the uncomputable ones. It is in particular true in the games models, but it is of course not satisfied by the term model of μ PCF, and this is the main motivation for working in a model of μ PCF rather than directly with the syntactic language.

Assuming R^C satisfies these two assumptions, we now prove that the interpretation of (DC^r) realizes (DC^r):

Lemma 3.17. *Let $\perp \subseteq R^{\mathcal{C}}(\mathbf{1}, [I])$.*

$$M_{(\mathbf{DC}^r)} = \lambda xy. \text{barrec} (\lambda v. x \mid v \mid (\text{if}_0 \mid v \mid \bar{0} (\pi_1 v \mid \text{pred} \mid v \mid \downarrow))) y \epsilon \in \left| \forall \vec{u}^{\vec{r}} (\forall^r x^t \forall^r y^\sigma (\forall z^\sigma \neg A \{x, y, z\} \Rightarrow \forall x'^t A \{x', y, y\}) \Rightarrow \exists v'^t \rightarrow^\sigma \forall^r x^t A \{x, v x, v (\mathbf{S} x)\}) \right|$$

where A denotes a formula over $\Sigma_{CA^{\omega^r}} = \Sigma_{PA^{\omega^r}}$ with free variables among $x^t, y^\sigma, z^\sigma, \vec{u}^{\vec{r}}$ and which is of the shape $(\downarrow z) \wedge B$. For clarity, we write $A \{t, u, v\}$ instead of $A \{t/x, u/y, v/z\}$.

Proof. Recall that we use the bar recursor:

$$\text{barrec} : (A^{*\diamond} \rightarrow (A^* \rightarrow 0) \rightarrow A^*) \rightarrow ((I \rightarrow A^*) \rightarrow 0) \rightarrow A^{*\diamond} \rightarrow 0$$

To simplify notations we define:

$$\text{last} \triangleq \lambda v. \text{if}_0 \mid v \mid \bar{0} (\pi_1 v \mid \text{pred} \mid v \mid \downarrow)$$

so $M_{(\mathbf{DC}^r)}$ is:

$$M_{(\mathbf{DC}^r)} = \lambda xy. \text{barrec} (\lambda v. x \mid v \mid (\text{last } v)) y \epsilon : (I \rightarrow T \rightarrow (A^* \rightarrow 0) \rightarrow A^*) \rightarrow ((I \rightarrow A^*) \rightarrow 0) \rightarrow 0$$

Let now $\vec{a} \in \vec{r}^{\mathcal{M}}$ and write $A' \equiv A \{\vec{a}/\vec{u}\}$. We use again the notation $A' \{t, u, v\}$. Let:

$$\phi \in R^{\mathcal{C}}(\mathbf{1}, [I \rightarrow T \rightarrow (A^* \rightarrow 0) \rightarrow A^*] \wp [I]) \quad \psi \in R^{\mathcal{C}}(\mathbf{1}, [(I \rightarrow A^*) \rightarrow 0] \wp [I])$$

be such that:

$$\phi \in \left| \forall^r x^t \forall^r y^\sigma (\forall z^\sigma \neg A' \{x, y, z\} \Rightarrow \forall x'^t A' \{x', y, y\}) \right| \quad \psi \in \left| \forall v'^t \rightarrow^\sigma \neg \forall^r x^t A' \{x, v x, v (\mathbf{S} x)\} \right|$$

We then have to prove:

$$\text{barrec} (\lambda v. \phi \mid v \mid (\text{last } v)) \psi \epsilon \in \perp$$

For conciseness we write:

$$\zeta = \text{barrec} (\lambda v. \phi \mid v \mid (\text{last } v)) \psi \epsilon \in R^{\mathcal{C}}(\mathbf{1}, [A^{*\diamond} \rightarrow 0] \wp [I])$$

so we must prove that $\zeta \epsilon \in \perp$. With our notations, for any $\xi \in R^{\mathcal{C}}(\mathbf{1}, [A^{*\diamond}] \wp [I])$ we have:

$$\begin{aligned} \zeta \xi &= \text{barrec} (\lambda v. \phi \mid v \mid (\text{last } v)) \psi \xi \\ &= \psi (\xi @ \phi \mid \xi \mid (\text{last } \xi) (\lambda x. \text{barrec} (\lambda v. \phi \mid v \mid (\text{last } v)) \psi (\xi * x))) \\ &= \psi (\xi @ \phi \mid \xi \mid (\text{last } \xi) (\lambda x. \zeta (\xi * x))) \end{aligned}$$

The following iteration lemma (the proof of which is deferred to the end of the section) is the heart of the adequacy:

Lemma 3.18. *Let $b_0, \dots, b_n \in \sigma^{\mathcal{M}}$ and $\varphi_0, \dots, \varphi_{n-1} \in R^{\mathcal{C}}(\mathbf{1}, [A^*] \wp [I])$ be such that:*

$$b_0 = 0^{\sigma^{\mathcal{M}}} \quad \forall 0 \leq i < n, \varphi_i \in \left| A' \{i, b_i, b_{i+1}\} \right| \quad \zeta (\varphi_0 * \dots * \varphi_{n-1}) \notin \perp$$

where 0^σ is the term from section 1.3. There exists $b_{n+1} \in \sigma^{\mathcal{M}}$ and $\varphi_n \in R^{\mathcal{C}}(\mathbf{1}, [A^*] \wp [I])$ such that:

$$\varphi_n \in \left| A' \{n, b_n, b_{n+1}\} \right| \quad \zeta (\varphi_0 * \dots * \varphi_{n-1} * \varphi_n) \notin \perp$$

In order to prove $\zeta \in |\perp|$, we use a reductio-ad-absurdum and suppose $\zeta \notin |\perp|$. By iterating the lemma, we build sequences $(b_n)_{n \in \mathbb{N}}$ in $\sigma^{\mathcal{M}}$ and $(\varphi_n)_{n \in \mathbb{N}}$ in $R^{\mathcal{C}}(\mathbf{1}, [B^*] \wp [I])$ such that:

$$b_0 = 0^{\sigma^{\mathcal{M}}} \quad \forall n \in \mathbb{N}, \varphi_n \in |A' \{n, b_n, b_{n+1}\}| \text{ and } \zeta(\varphi_0 * \dots * \varphi_{n-1}) \notin |\perp|$$

Since \mathcal{M} satisfies **(DC)**, we can build $b \in (\iota \rightarrow \sigma)^{\mathcal{M}}$ such that for every $n \in \mathbb{N}$, $b n = b_n$, and using the second assumption on $R^{\mathcal{C}}$ (definition 3.16) we also build $\varphi \in R^{\mathcal{C}}(\mathbf{1}, [I \rightarrow A^*] \wp [I])$ such that for every $n \in \mathbb{N}$, $\varphi \bar{n} = \varphi_n$. We now prove that:

$$\varphi \in |\forall^r x^t A' \{x, b x, b(\mathbf{S} x)\}|$$

Since $|\langle c \rangle| = \emptyset$ if $c \notin \mathbb{N}$, it is sufficient to prove that for $n \in \mathbb{N}$, $\varphi \bar{n} \in |A' \{n, b n, b(n+1)\}|$, but since $\varphi \bar{n} = \varphi_n$, $b n = b_n$ and $b(n+1) = b_{n+1}$, this is immediate. Now, since:

$$\psi \in |\forall v^{\iota \rightarrow \sigma} \neg \forall^r x^t A' \{x, v x, v(\mathbf{S} x)\}| \subseteq |\neg \forall^r x^t A' \{x, b x, b(\mathbf{S} x)\}|$$

we get $\psi \varphi \in |\perp|$, so $\mu\kappa.\psi \varphi \in \perp$. We use now the first assumption on $R^{\mathcal{C}}$ (definition 3.15), so there is some $m \in \mathbb{N}$ such that any morphism $\varphi' \in R^{\mathcal{C}}(\mathbf{1}, [I \rightarrow B^*] \wp [I])$ which satisfies:

$$\forall m' < m, \varphi' \bar{m}' = \varphi \bar{m}'$$

is such that $\mu\kappa.\psi \varphi' = \mu\kappa.\psi \varphi \in \perp$. If we write $\xi = \varphi_0 * \dots * \varphi_{m-1}$, this is verified in particular for:

$$\xi @ \phi |\xi| (\text{last } \xi) (\lambda x. \zeta (\xi * x))$$

so we get:

$$\mu\kappa.\psi (\xi @ \phi |\xi| (\text{last } \xi) (\lambda x. \zeta (\xi * x))) \in \perp$$

and finally:

$$\zeta (\varphi_0 * \dots * \varphi_{m-1}) = \zeta \xi = \psi (\xi @ \phi |\xi| (\text{last } \xi) (\lambda x. \zeta (\xi * x))) \in |\perp|$$

from which we get our contradiction. \square

Here is the proof of the iteration lemma:

Proof. Write $\xi = \varphi_0 * \dots * \varphi_{n-1}$. We have:

$$\psi (\xi @ \phi |\xi| (\text{last } \xi) (\lambda x. \zeta (\xi * x))) = \zeta \xi = \zeta (\varphi_0 * \dots * \varphi_{n-1}) \notin |\perp|$$

Using $\mathbf{k}^{\mathcal{M}}$, $\mathbf{s}^{\mathcal{M}}$ and $\mathbf{rec}^{\mathcal{M}}$ we can build some $b \in (\iota \rightarrow \sigma)^{\mathcal{M}}$ such that for any $0 \leq i \leq n$, $b i = b_i$, and for any $i > n$, $b i = b_n$. Since $\psi \in |\forall v^{\iota \rightarrow \sigma} \neg \forall^r x^t A' \{x, v x, v(\mathbf{S} x)\}|$, we have in particular:

$$\psi \in |\neg \forall^r x^t A' \{x, b x, b(\mathbf{S} x)\}|$$

therefore:

$$\xi @ \phi |\xi| (\text{last } \xi) (\lambda x. \zeta (\xi * x)) \notin |\forall^r x^t A' \{x, b x, b(\mathbf{S} x)\}|$$

Since $|\langle c \rangle| = \emptyset$ if $c \notin \mathbb{N}$, there must be some $i \in \mathbb{N}$ such that:

$$(\xi @ \phi |\xi| (\text{last } \xi) (\lambda x. \zeta (\xi * x))) \bar{i} \notin |A' \{i, b_i, b_{i+1}\}|$$

If $i < n$ then we get:

$$\xi \bar{i} = \varphi_i \notin |A' \{i, b_i, b_{i+1}\}|$$

which contradicts the hypothesis of the lemma. Therefore, $i \geq n$ and:

$$\phi |\xi| (\text{last } \xi) (\lambda x. \zeta (\xi * x)) \notin |A' \{i, b_i, b_{i+1}\}| = |A' \{i, b_n, b_n\}|$$

First, since $\xi = \varphi_0 * \dots * \varphi_{n-1}$ we have $|\xi| = \bar{n} \in |\langle n \rangle|$, and therefore:

$$\phi |\xi| \in |\forall^r y^\sigma (\forall z^\sigma \neg A' \{n, y, z\} \Rightarrow \forall x'^\iota A' \{x', y, y\})|$$

We prove now that $\text{last } \xi \in |\langle b_n \rangle|$ by distinguishing cases:

- $n = 0$: $\text{last } \xi = \text{if}_0 |\xi| \bar{0} (\pi_1 \xi \downarrow \text{pred } |\xi|) = \bar{0} \in |\langle 0^{\sigma^{\mathcal{M}}} \rangle| = |\langle b_0 \rangle|$ (indeed, $\bar{0} \in |\langle 0^{\sigma^{\mathcal{M}}} \rangle|$ by an easy induction on σ)
- $n \neq 0$: $\text{last } \xi = \text{if}_0 |\xi| \bar{0} (\pi_1 \xi \downarrow \text{pred } |\xi|) = \pi_1 \varphi_{n-1} \in |\langle b_n \rangle|$ since:

$$\varphi_{n-1} \in |A' \{n-1, b_{n-1}, b_n\}| = |\langle b_n \rangle \wedge B \{\bar{a}/\bar{u}, n-1/x, b_{n-1}/y, b_n/z\}|$$

Therefore we have:

$$\phi |\xi| (\text{last } \xi) \in |\forall z^\sigma \neg A' \{n, b_n, z\} \Rightarrow \forall x'^\iota A' \{x', b_n, b_n\}|$$

and so:

$$\lambda x. \zeta (\xi * x) \notin |\forall z^\sigma \neg A' \{n, b_n, z\}|$$

which means that there exists some $b_{n+1} \in \sigma^{\mathcal{M}}$ such that:

$$\lambda x. \zeta (\xi * x) \notin |\neg A' \{n, b_n, b_{n+1}\}|$$

and so there exists $\varphi_n \in R^{\mathcal{C}}(\mathbf{1}, [A^*] \wp [I])$ such that $\varphi_n \in |A' \{n, b_n, b_{n+1}\}|$ and:

$$\zeta (\varphi_0 * \dots * \varphi_{n-1} * \varphi_n) = \zeta (\xi * \varphi_n) = (\lambda x. \zeta (\xi * x)) \varphi_n \notin |\perp| \quad \square$$

3.4. Extraction. Since the Friedman translation is directly built in the realizability interpretation, the extraction result is an easy consequence of the definitions. We show that from any Π_2^0 -formula provable in CA^ω we can extract a computable witness in $R^{\mathcal{C}}$. Note that in the extraction lemma, the equality $t =_\tau u$ is at any type:

Lemma 3.19. *From a proof of $\vdash \forall x^\sigma \exists y^\iota (t =_\tau u)$ in CA^ω , one can extract a $\lambda\mu$ -term $M : T \rightarrow I$ such that for any $a \in \sigma^{\mathcal{M}}$ and $\phi \in |\langle a \rangle|$, there is some $n \in \mathbb{N}$ such that:*

$$(t \{a/x, n/y\})^{\mathcal{M}} = (u \{a/x, n/y\})^{\mathcal{M}} \text{ and } M \phi = \bar{n} \text{ in } R^{\mathcal{C}}$$

Proof. Since \exists and $=$ are just encodings, we actually have a proof of $\vdash \forall x^\sigma \neg \forall y^\iota \neg (t \neq_\tau u)$. First, we can easily turn it into a proof of $\vdash \forall x^\sigma \neg \forall y^\iota (t \neq_\tau u)$, and we get by relativization a proof \mathfrak{p} in CA^{ω^r} of $\vdash \forall^r x^\sigma \neg \forall^r y^\iota (t \neq_\tau u)$. The adequacy lemma then tells us:

$$\mathfrak{p}^* \in |\forall^r x^\sigma \neg \forall^r y^\iota (t \neq_\tau u)|$$

If $a \in \sigma^{\mathcal{M}}$ and $\phi \in |\langle a \rangle|$, we get $\mathfrak{p}^* \phi \in |\neg \forall^r y^\iota (t \{a/x\} \neq_\tau u \{a/x\})|$. Let fix now:

$$\perp = \left\{ \bar{n} \in R^{\mathcal{C}}(\mathbf{1}, [I]) \mid (t \{a/x, n/y\})^{\mathcal{M}} = (u \{a/x, n/y\})^{\mathcal{M}} \right\}$$

We prove that $\lambda x. [\kappa] x \in |\forall^r y^\iota (t \{a/x\} \neq_\tau u \{a/x\})|$. For that let $n \in \iota^{\mathcal{M}}$, and let prove that $(\lambda x. [\kappa] x) \bar{n} = [\kappa] \bar{n} \in |t \{a/x, n/y\} \neq u \{a/x, n/y\}|$. There are two cases:

- $\bar{n} \in \perp$: in that case, $(t \{a/x, n/y\})^{\mathcal{M}} = (u \{a/x, n/y\})^{\mathcal{M}}$, so:

$$|t \{a/x, n/y\} \neq u \{a/x, n/y\}| = |\perp|$$

and $\mu\kappa. [\kappa] \bar{n} = \bar{n} \in \perp$ so $[\kappa] \bar{n} \in |\perp|$

- $\bar{n} \notin \perp$: in that case, $(t \{a/x, n/y\})^{\mathcal{M}} \neq (u \{a/x, n/y\})^{\mathcal{M}}$, so:

$$|t \{a/x, n/y\} \neq u \{a/x, n/y\}| = R^{\mathcal{C}}(\mathbf{1}, [0] \wp [I])$$

and therefore $[\kappa] \bar{n} \in |t \{a/x, n/y\} \neq u \{a/x, n/y\}|$

We get $\mathfrak{p}^* \phi (\lambda x. [\kappa] x) \in |\perp|$, so $\mu\kappa. \mathfrak{p}^* \phi (\lambda x. [\kappa] x) \in \perp$. This means that $\mu\kappa. \mathfrak{p}^* \phi (\lambda x. [\kappa] x)$ is some $\bar{n} \in \perp$ (so $(t \{a/x, n/y\})^{\mathcal{M}} = (u \{a/x, n/y\})^{\mathcal{M}}$). Finally, we have the claimed result with $M \triangleq \lambda u. \mu\kappa. \mathfrak{p}^* u (\lambda x. [\kappa] x)$. \square

Similarly to what was done in [BR13], we can define an operational semantics for μ PCF and adapt the techniques of e.g. [AC98] to prove computational adequacy with respect to a non-degenerated R^C , i.e. if $M : \iota$ in μ PCF, then for any $n \in \mathbb{N}$:

$$M \rightarrow^* \bar{n} \iff M = \bar{n} \text{ in } R^C$$

Using computational adequacy and the extraction lemma in the particular case of $\sigma = \iota$, we get that for any $m \in \mathbb{N}$, there is some $n \in \mathbb{N}$ such that $M \bar{n} \rightarrow^* \bar{n}$ and $(t\{m/x, n/y\})^M = (u\{m/x, n/y\})^M$.

CONCLUSION

Realizability interpretation of classical logic with control operators is a recent field in which much of the contributions take place in Krivine's untyped setting. We defined here a realizability model in which proofs are interpreted in a category of continuations, which is the universal model of typed $\lambda\mu$ -calculus. The duality between terms and contexts, which appears in Krivine's work as a duality between terms and stacks, is reflected here by the duality in a category of continuations R^C between \mathcal{C} and R^C .

The direct interpretations of classical logic can be seen as a way to avoid Gödel's negative translation on formulas, by using programming languages with an operational semantics which corresponds to that of their CPS-translation to λ -calculus. Similarly, the choice of having a free μ -variable κ in our realizers may be seen as a way to avoid Friedman's A -translation. Indeed, in Friedman's original work, the translation is obtained by replacing each basic predicate P with the disjunction $P \vee A$. In classical sequent calculus, the right-hand context is meant to be interpreted as a disjunction, so adding a fixed μ -variable to this context corresponds to applying Friedman's translation on programs instead of proofs. In many implementations of Friedman's trick, the formula A is not added as a disjunction to every predicate, but instead the \perp formula is replaced everywhere with A . Our implementation of Friedman's translation however allows simpler interpretations of the realizers in some models. Indeed, the continuation passing hidden behind the usual implementations is abstracted with the help of $\lambda\mu$ -calculus constructs. Rather than adding a continuation on top of every interpretation of \perp , we only add one continuation on top of the whole realizer. This simplicity of the interpretation requires some properties on the model, and in particular it rules out Scott domains. However, all these properties are satisfied in unbracketed games, and it would be interesting to find other models satisfying these. The natural candidates for this are bistable biorders [Lai07] and sequential algorithms [BC82], but these are not the only ones. Coherent spaces may also satisfy these properties if we choose carefully the object of natural numbers.

The μ -variable κ is also exploited in the definition of the realizability interpretation, to parameterize the orthogonality relation between realizers and counter-realizers. However, contrary to usual interpretations of classical logic, we consider positive predicates, for which no counter-realizers are defined. Through a suitable restriction on proofs we preserve the adequacy of the interpretation. Roughly, it amounts to forbid classical reasoning on the positive formulas, while keeping it in the general case. This introduction of positive predicates in a classical, negative setting is mainly motivated by the decomposition of the relativized universal quantifier into a uniform quantifier and a relativization predicate, the relativization predicate being a positive one.

We proved that the usual terms of Gödel’s system T realize the axioms of Peano arithmetic in our direct setting, without CPS-translation. This is not a surprise since all these axioms do imply their negative translation intuitionistically. However, when it comes to the axiom of choice, this is not as easy and we prove, as was done in [BR13], that the bar recursion operator realizes it in a direct interpretation. Interestingly, this bar recursor also gives computational content to the double-negation shift principle in an intuitionistic setting, and that same principle allows to derive intuitionistically the negative translation of the axiom of choice from the axiom of choice.

Finally, we validate our model by proving an extraction result for Π_2^0 formulas, relying once again on the μ -variable κ which allows the orthogonality relation to be a parameter of the model.

Acknowledgments. I thank the reviewers for their careful reading of my work, which led to constructive comments and suggestions.

REFERENCES

- [AC98] Roberto Amadio and Pierre-Louis Curien. *Domains and Lambda-Calculi*, volume 46 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1998.
- [AHS07] Zena Ariola, Hugo Herbelin, and Amr Sabry. A proof-theoretic foundation of abortive continuations. *Higher-Order and Symbolic Computation*, 20(4):403–429, 2007.
- [BBC98] Stefano Berardi, Marc Bezem, and Thierry Coquand. On the Computational Content of the Axiom of Choice. *Journal of Symbolic Logic*, 63(2):600–622, 1998.
- [BC82] Gérard Berry and Pierre-Louis Curien. Sequential Algorithms on Concrete Data Structures. *Theoretical Computer Science*, 20(3):265–321, 1982.
- [Blo14] Valentin Blot. *Game semantics and realizability for classical logic*. PhD thesis, École Normale Supérieure de Lyon, 2014.
- [BO05] Ulrich Berger and Paulo Oliva. Modified bar recursion and classical dependent choice. In *Logic Colloquium ’01, Proceedings of the Annual European Summer Meeting of the Association for Symbolic Logic*, volume 20 of *Lecture Notes in Logic*, pages 89–107. A K Peters, Ltd., 2005.
- [BR13] Valentin Blot and Colin Riba. On Bar Recursion and Choice in a Classical Setting. In *11th Asian Symposium on Programming Languages and Systems*, volume 8301 of *Lecture Notes in Computer Science*, pages 349–364. Springer, 2013.
- [CH00] Pierre-Louis Curien and Hugo Herbelin. The duality of computation. In *5th International Conference on Functional Programming*, pages 233–243. ACM Press, 2000.
- [dG94] Philippe de Groote. On the Relation between the Lambda-Mu-Calculus and the Syntactic Theory of Sequential Control. In *5th International Conference on Logic Programming and Automated Reasoning*, volume 822 of *Lecture Notes in Computer Science*, pages 31–43. Springer, 1994.
- [DP01] René David and Walter Py. $\lambda\mu$ -calculus and Böhm’s theorem. *Journal of Symbolic Logic*, 66(1):407–413, 2001.
- [EO12] Martín Hötzel Escardó and Paulo Oliva. The Peirce translation. *Annals of Pure and Applied Logic*, 163(6):681–692, 2012.
- [Gir91] Jean-Yves Girard. A New Constructive Logic: Classical Logic. *Mathematical Structures in Computer Science*, 1(3):255–296, 1991.
- [Gir93] Jean-Yves Girard. On the Unity of Logic. *Annals of Pure and Applied Logic*, 59(3):201–217, 1993.
- [Göd33] Kurt Gödel. Zur intuitionistischen Arithmetik und Zahlentheorie. *Ergebnisse eines mathematischen Kolloquiums*, 4:34–38, 1933.
- [Göd58] Kurt Gödel. Über eine bisher noch nicht benützte Erweiterung des finiten Standpunktes. *Dialectica*, 12(3-4):280–287, 1958.
- [Gri90] Timothy Griffin. A Formulae-as-Types Notion of Control. In *17th Symposium on Principles of Programming Languages*, pages 47–58. ACM Press, 1990.

- [Her95] Hugo Herbelin. *Séquents qu'on calcule: de l'interprétation du calcul des séquents comme calcul de λ -termes et comme calcul de stratégies gagnantes*. PhD thesis, Université Paris 7, 1995.
- [HO00] Martin Hyland and Luke Ong. On Full Abstraction for PCF: I, II, and III. *Information and Computation*, 163(2):285–408, 2000.
- [HS02] Martin Hofmann and Thomas Streicher. Completeness of Continuation Models for $\lambda\mu$ -Calculus. *Information and Computation*, 179(2):332–355, 2002.
- [HS09] Hugo Herbelin and Alexis Saurin. $\lambda\mu$ -calculus and $\Lambda\mu$ -calculus: a Capital Difference. <http://hal.inria.fr/inria-00524942>, 2009.
- [Kle45] Stephen Cole Kleene. On the Interpretation of Intuitionistic Number Theory. *Journal of Symbolic Logic*, 10(4):109–124, 1945.
- [Koh90] Ulrich Kohlenbach. *Theory of majorizable and continuous functionals and their use for the extraction of bounds from non-constructive proofs: effective moduli of uniqueness for best approximations from ineffective proofs of uniqueness*. PhD thesis, Goethe Universität Frankfurt, 1990.
- [Koh08] Ulrich Kohlenbach. *Applied Proof Theory: Proof Interpretations and their Use in Mathematics*. Springer Monographs in Mathematics. Springer, 2008.
- [Kre59] Georg Kreisel. Interpretation of analysis by means of constructive functionals of finite types. In *Constructivity in mathematics: Proceedings of the colloquium held at Amsterdam, 1957*, Studies in Logic and the Foundations of Mathematics, pages 101–128. North-Holland Publishing Company, 1959.
- [Kri94] Jean-Louis Krivine. A General Storage Theorem for Integers in Call-by-Name lambda-Calculus. *Theoretical Computer Science*, 129(1):79–94, 1994.
- [Kri01] Jean-Louis Krivine. Typed lambda-calculus in classical Zermelo-Frænkel set theory. *Archive for Mathematical Logic*, 40(3):189–205, 2001.
- [Kri03] Jean-Louis Krivine. Dependent choice, 'quote' and the clock. *Theoretical Computer Science*, 308(1–3):259–276, 2003.
- [Kri09] Jean-Louis Krivine. Realizability in classical logic. *Panoramas et synthèses*, 27:197–229, 2009.
- [Lai97] James Laird. Full Abstraction for Functional Languages with Control. In *12th Annual IEEE Symposium on Logic in Computer Science*, pages 58–67. IEEE Computer Society, 1997.
- [Lai99] James Laird. *A semantic analysis of control*. PhD thesis, University of Edinburgh, 1999.
- [Lai07] James Laird. Bistable Borders: A Sequential Domain Theory. *Logical Methods in Computer Science*, 3(2), 2007.
- [LRS93] Yves Lafont, Bernhard Reus, and Thomas Streicher. Continuations Semantics or Expressing Implication by Negation. Technical Report 93-21, Ludwig-Maximilians-Universität, München, 1993.
- [Miq11] Alexandre Miquel. Existential witness extraction in classical realizability and via a negative translation. *Logical Methods in Computer Science*, 7(2), 2011.
- [Ong96] Luke Ong. A Semantic View of Classical Proofs: Type-Theoretic, Categorical, and Denotational Characterizations (Preliminary Extended Abstract). In *11th Annual IEEE Symposium on Logic in Computer Science*, pages 230–241. IEEE Computer Society, 1996.
- [OS97] Luke Ong and Charles Stewart. A Curry-Howard Foundation for Functional Computation with Control. In *24th Symposium on Principles of Programming Languages*, pages 215–227. ACM Press, 1997.
- [OS08] Paulo Oliva and Thomas Streicher. On Krivine's Realizability Interpretation of Classical Second-Order Arithmetic. *Fundamenta Informaticae*, 84(2):207–220, 2008.
- [Par92] Michel Parigot. $\lambda\mu$ -Calculus: An Algorithmic Interpretation of Classical Natural Deduction. In *3rd International Conference on Logic Programming and Automated Reasoning*, volume 624 of *Lecture Notes in Computer Science*, pages 190–201. Springer, 1992.
- [Plo77] Gordon Plotkin. LCF Considered as a Programming Language. *Theoretical Computer Science*, 5(3):223–255, 1977.
- [Pow13] Thomas Powell. *On bar recursive interpretations of analysis*. PhD thesis, Queen Mary University of London, 2013.
- [Pow14] Thomas Powell. The equivalence of bar recursion and open recursion. *Annals of Pure and Applied Logic*, 165(11):1727–1754, 2014.
- [Sau05] Alexis Saurin. Separation with Streams in the $\Lambda\mu$ -calculus. In *20th IEEE Symposium on Logic in Computer Science*, pages 356–365. IEEE Computer Society, 2005.

- [Sco93] Dana Scott. A Type-Theoretical Alternative to ISWIM, CUCH, OWHY. *Theoretical Computer Science*, 121(1-2):411–440, 1993.
- [Sel01] Peter Selinger. Control categories and duality: on the categorical semantics of the $\lambda\mu$ calculus. *Mathematical Structures in Computer Science*, 11(2):207–260, 2001.
- [Spe62] Clifford Spector. Provably recursive functionals of analysis: a consistency proof of analysis by an extension of principles in current intuitionistic mathematics. In *Recursive Function Theory: Proceedings of Symposia in Pure Mathematics*, volume 5, pages 1–27. American Mathematical Society, 1962.
- [TVD88] Anne Sjerp Troelstra and Dirk Van Dalen. *Constructivism in mathematics: an introduction*, volume 121, 123 of *Studies in logic and the foundations of mathematics*. Elsevier, 1988.