Hybrid realizability for intuitionistic and classical choice

Valentin Blot

Hybrid realizability for intuitionistic and classical choice

Valentin Blot
University of Bath
v.blot@bath.ac.uk

Abstract

In intuitionistic realizability like Kleene's or Kreisel's, the axiom of choice is trivially realized. It is even provable in Martin-Löf's intuitionistic type theory. In classical logic, however, even the weaker axiom of countable choice proves the existence of non-computable functions. This logical strength comes at the price of a complicated computational interpretation which involves strong recursion schemes like bar recursion. We take the best from both worlds and define a realizability model for arithmetic and the axiom of choice which encompasses both intuitionistic and classical reasoning. In this model two versions of the axiom of choice can co-exist in a single proof: intuitionistic choice and classical countable choice. We interpret intuitionistic choice efficiently, however its premise cannot come from classical reasoning. Conversely, our version of classical choice is valid in full classical logic, but it is restricted to the countable case and its realizer involves bar recursion. Having both versions allows us to obtain efficient extracted programs while keeping the provability strength of classical logic.

Categories and Subject Descriptors F.3.2 [*Semantics of Programming Languages*]: Denotational semantics; F.4.1 [*Mathematical Logic*]

1. Introduction

Realizability appeared in [16] as a formal account of the Brouwer-Heyting-Kolmogorov interpretation of logic, leading to the Curry-Howard isomorphism between intuitionistic proofs and purely functional programs. In [11], Griffin used control operators to extend the isomorphism to classical logic. While the first realizability interpretations of classical logic relied on a negative translation followed by an intuitionistic realizability interpretation, Griffin's discovery can be exploited to give a direct realizability interpretation of classical logic. This allowed Krivine to interpret second-order Peano arithmetic and the axiom of dependent choice in an untyped λ-calculus extended with the `call/cc` operator [17]. In the work presented here, we interpret first-order classical arithmetic and the axiom of countable choice in a model of the simply-typed $\lambda\mu$-calculus [20], an extension of λ-calculus with control features.

The axiom of choice is ubiquitous in mathematics and is often used without even noticing. Therefore, having a computational interpretation of this axiom is essential to the extraction of programs from a wide range of mathematical proofs. While in usual interpretations of intuitionistic logic the axiom of choice is trivially realized, interpreting even its countable version in a classical setting requires the use of strong recursion schemes, like the bar recursion operator defined in [23] in the framework of Gödel's Dialectica interpretation. The requirement for such a strong recursion principle can be explained by the much stronger provability strength of classical choice. For instance, since any formula can be reflected by a boolean in classical logic, the axiom of countable choice can build the characteristic function of any formula with integer parameters. In particular, one can choose formulas encoding non-decidable predicates and prove the existence of non-computable functions.

Variants of bar recursion were used in [2, 3] to interpret the negative translation of the axiom of choice in an intuitionistic setting, and therefore classical arithmetic with countable choice through Gödel's negative translation. In [8], it was shown that bar recursion can be used in a language with control operators to interpret directly the axiom of countable choice in a classical setting. In the present work we extend this approach, adding strong existentials to the realizability interpretation. While these strong existentials are known to raise issues in the presence of control operators [12], our proof system allows for a simple criterion which forbids classical reasoning on these, ensuring the correctness of our computational interpretation in $\lambda\mu$-calculus. Conversely, weak existentials (which, in our setting, are double negations of the strong ones) work well with control operators, but are less efficient from the computational perspective because their interpretations can hide some backtracks.

Rather than having to choose between an efficient system which is restricted to intuitionistic logic or a full classical system with a complex computational interpretation, we take the best from both worlds and work within classical logic with strong existential quantifications, using their weak counterparts when classical reasoning is needed. In a proof where the excluded middle is never used on some existential formula, this existential can be strong and benefit from an efficient interpretation (in particular, the axiom of choice is trivially realized in that case). If in the same proof some classical reasoning is performed on another existential formula, then that existential must be weak, and while we can still use the axiom of countable choice on it, its computational interpretation is given by bar recursion and can involve a costly recursion.

We validate our model with an extraction result from proofs of Π^0_2 formulas with either a strong or a weak existential. The case of a strong existential is immediate by definition of its semantics, while the case of a weak existential relies on Friedman's trick which, in direct interpretations of classical logic, amounts to a non-empty realizability interpretation of the false formula. Our system allows for extraction of more efficient programs than with the usual direct or indirect interpretations of classical logic, provided some care is taken to choose strong existentials whenever possible.

The combination of strong existentials and classical logic was also investigated in [13], where strong existentials were weakened enough to make them compatible with classical logic while keeping

LICS '16, July 05-08, 2016, New York, NY, USA
ACM 978-1-4503-4391-6/16/07.
http://dx.doi.org/10.1145/2933575.2934511

the validity of countable and dependent choices. The goal of [13] was to find the minimal restriction of strong existentials which makes them compatible with control operators. This was achieved by an insightful analysis of proofs leading to the definition of the *negative-elimination-free* ones, but came at the expense of a complex computational interpretation, similarly to what happens with bar recursion. In the present work, we have a different goal and achieve it by following a different path: we keep the full power of strong existentials (and therefore their simple computational interpretation) and use bar recursion to computationally interpret the classical axiom of countable choice with weak existentials. In our goal of extraction of efficient programs, the presence of strong existentials with their full power and simple interpretation provides more efficient computational interpretations of classical proofs than [13].

In section 2 we define the programming language System μT_{br} in which proofs of our system are interpreted, and present the categorical models of this language. In section 3 we present our deduction system, which is a hybrid system allowing both intuitionistic and classical reasoning. We discuss the provability strength of this system and in particular how it can interpret both intuitionistic arithmetic with choice and classical arithmetic with countable choice. Then we prove that the terms interpreting our proofs are well-typed in System μT_{br}. Finally, we present our realizability interpretation in section 4, we prove its adequacy for our deduction system, and we validate our computational interpretation with two extraction results: for strong and weak existentials.

2. $\lambda\mu$-calculus and categories of continuations

In this section we define the programming language in which the proofs will be interpreted. This language is an extension of Gödel's system T with the control features of $\lambda\mu$-calculus [20] and bar recursion. We call this language System μT_{br}. In order to have adequacy of the bar recursion operator for our variant of double-negation shift, we do not work directly in System μT_{br} but in a model of it. We therefore recall the definition of categories of continuations [14], which are the universal models of $\lambda\mu$-calculus.

2.1 System μT_{br}

$\lambda\mu$-calculus has been introduced in [20] to provide a Curry-Howard correspondence for classical natural deduction. We use here a particular $\lambda\mu$-theory with natural numbers and recursion that we call System μT. The types of System μT are built from a single base type for natural numbers, the empty type, product and arrow types:

$$T, U \quad ::= \quad I \mid 0 \mid T \times U \mid T \to U$$

The terms are given along with the typing rules in figure 1, and the set $\mathcal{C}st$ of constants for System μT are:

$$\overline{\mathsf{n}} : I \qquad \mathsf{succ} : I \to I \qquad \mathsf{rec} : T \to (I \to T \to T) \to I \to T$$

The equational theory of System μT is given in figure 2, where both sides of the equations must be well-typed.

In order to define the bar recursion operator, we need to be able to manipulate finite lists of elements. There are several ways to implement these in System μT. The particular implementation chosen is irrelevant to our interpretation, so we fix one particular encoding and write $T^\diamond$ for the type of lists of elements of type T. We also define some notations for the operations on finite lists: we write $\epsilon : T^\diamond$ for the empty list, $M * N : T^\diamond$ for the list obtained by appending $N : T$ to $M : T^\diamond$, $|M| : I$ for the size of $M : T^\diamond$ and $M \mathbin{@} N : I \to T$ for the infinite extension of $M : T^\diamond$ with the values from $N : I \to T$, i.e.:

$$(\epsilon * M_0 * \ldots * M_{n-1} \mathbin{@} N)\,\overline{\mathsf{m}} = \begin{cases} M_m & \text{if } m < n \\ N\,\overline{\mathsf{m}} & \text{otherwise} \end{cases}$$

Given this fixed implementation of lists in System μT, we can now extend it to System μT_{br} by adding the bar recursion operator:

$$\mathsf{brec} : (I \times (T \to 0) \to I \to T) \to ((I \to T) \to 0) \to T^\diamond \to 0$$

together with its defining equation:

$$\mathsf{brec}\, M\, N\, P = N\,(P \mathbin{@} M\,\langle |P|\,, \lambda x.\mathsf{brec}\, M\, N\,(P * x)\rangle)$$

We choose to present System μT_{br} as an equational theory for simplicity, but one can define an abstract machine (as was done in [8]) for which computational adequacy holds: for any term $M : I$ of System μT_{br} and any $n \in \mathbb{N}$, $M = \overline{\mathsf{n}}$ in the equational theory if and only if M reduces to $\overline{\mathsf{n}}$ in the abstract machine.

2.2 Categories of continuations

As stated in the introduction, the axiom of countable choice together with classical logic proves the existence of non-computable functions. Therefore, our language of realizers needs to contain such functions. While in [2] this was achieved by extending the language with infinite terms, we work as in [3] in a model of our programming language. Just like cartesian closed categories are the universal models of λ-calculus, categories of continuations [14] are the universal models of $\lambda\mu$-calculus. A category of continuations is a full subcategory of a distributive category with exponentials of a fixed object:

Definition 1 (Category of continuations)**.** *Let $\mathcal{C}$ be a distributive category and let R be an object of $\mathcal{C}$ such that all exponentials R^A exist. Then the full subcategory $R^{\mathcal{C}}$ consisting of the objects R^A is called a category of continuations.*

We now fix a category of continuations $R^{\mathcal{C}}$ together with an object $[\![I]\!] \in Ob\,(\mathcal{C})$ and describe the semantics of System μT_{br} in $R^{\mathcal{C}}$. The object $[\![I]\!]$ is the negative interpretation of the type I, and from that parameter every type of System μT_{br} is given both a negative interpretation $[\![T]\!] \in Ob\,(\mathcal{C})$ and a positive one $[T] \triangleq R^{[\![T]\!]} \in Ob\left(R^{\mathcal{C}}\right)$ by:

$$[\![0]\!] \triangleq \mathbf{1} \qquad [\![T \times U]\!] \triangleq [\![T]\!] + [\![U]\!] \qquad [\![T \to U]\!] \triangleq [T] \times [\![U]\!]$$

where $\mathbf{1}$, $+$ and $\times$ correspond to the distributive structure of $\mathcal{C}$. We suppose a given interpretation $[\mathsf{c}] \in R^{\mathcal{C}}\,(\mathbf{1}, [T])$ for each $\mathsf{c} : T \in \mathcal{C}st$, from which we can define the interpretation of any term of System μT_{br}:

$$[x_1 : T_1, \ldots, x_n : T_n \vdash M : U \mid \alpha_1 : V_1, \ldots, \alpha_m : V_m] \\ \in R^{\mathcal{C}}\,([T_1] \times \ldots \times [T_n], [U] \parr ([V_1] \parr \ldots \parr [V_m]))$$

where $R^A \parr R^B \triangleq R^{A \times B}$ is a binoidal functor on $R^{\mathcal{C}}$ (see [22] for details).

We make some further assumptions on $R^{\mathcal{C}}$. First, the equations for the constants are verified in $R^{\mathcal{C}}$, and therefore the model is sound: if $M = N$ in System μT_{br}, then $[M] = [N]$ in $R^{\mathcal{C}}$. Second, the model is complete at type I: if $[M] = [N] \in R^{\mathcal{C}}\,(\mathbf{1}, [I])$, then $M = N$ in System μT_{br}. The model will further be required to satisfy sequence internalization (definition 2) and continuity (definition 3), but these requirements will be discussed in section 4.2, along with adequacy of bar recursion for classical choice.

We now briefly give two running examples of models of System μT_{br} satisfying soundness and adequacy at type I.

- The first one is a category of Scott domains. If we choose $\mathcal{C}$ to be the category of unpointed Scott domains (i.e. algebraic directed complete partial orders in which any two compatible elements have a lowest upper bound, see e.g. [1] for the definitions) and continuous functions, $R = \mathbb{N} \cup \{\bot\}$ with the ordering $a \leq b \Leftrightarrow a = b$ or $a = \bot$, which is the usual domain of natural numbers, and $[\![I]\!] = \{\bot\}$, then $R^{\mathcal{C}}$ is a category of

$$\frac{}{\vec{x} : \vec{T}, y : V \vdash y : V \mid \vec{\alpha} : \vec{U}} \qquad \frac{}{\vec{x} : \vec{T} \vdash \mathsf{c} : V \mid \vec{\alpha} : \vec{U}}\ (\mathsf{c}{:}V \in \mathcal{C}st)$$

$$\frac{\vec{x} : \vec{T}, y : V \vdash M : W \mid \vec{\alpha} : \vec{U}}{\vec{x} : \vec{T} \vdash \lambda y.M : V \to W \mid \vec{\alpha} : \vec{U}} \qquad \frac{\vec{x} : \vec{T} \vdash M : V \mid \vec{\alpha} : \vec{U} \qquad \vec{x} : \vec{T} \vdash N : W \mid \vec{\alpha} : \vec{U}}{\vec{x} : \vec{T} \vdash \langle M, N \rangle : V \times W \mid \vec{\alpha} : \vec{U}} \qquad \frac{\vec{x} : \vec{T} \vdash M : V \mid \beta : V, \vec{\alpha} : \vec{U}}{\vec{x} : \vec{T} \vdash [\beta]M : 0 \mid \beta : V, \vec{\alpha} : \vec{U}}$$

$$\frac{\vec{x} : \vec{T} \vdash M : V \to W \mid \vec{\alpha} : \vec{U} \qquad \vec{x} : \vec{T} \vdash N : V \mid \vec{\alpha} : \vec{U}}{\vec{x} : \vec{T} \vdash MN : W \mid \vec{\alpha} : \vec{U}} \qquad \frac{\vec{x} : \vec{T} \vdash M : V_1 \times V_2 \mid \vec{\alpha} : \vec{U}}{\vec{x} : \vec{T} \vdash \pi_i\, M : V_i \mid \vec{\alpha} : \vec{U}} \qquad \frac{\vec{x} : \vec{T} \vdash M : 0 \mid \beta : V, \vec{\alpha} : \vec{U}}{\vec{x} : \vec{T} \vdash \mu\beta.M : V \mid \vec{\alpha} : \vec{U}}$$

Figure 1. Typing rules for simply-typed $\lambda\mu$-calculus

$$(\lambda x.M)\, N = M\,\{N/x\} \qquad \lambda x.M\, x = M \quad (x \notin \mathrm{FV}(M)) \qquad (\mu\alpha.M)\, N = \mu\alpha.M\,\{[\alpha]\ \_\, N / [\alpha]\ \_\}$$
$$\pi_i\, \langle M_1, M_2 \rangle = M_i \qquad \langle \pi_1\, M, \pi_2\, M \rangle = M \qquad \pi_i\, (\mu\alpha.M) = \mu\alpha.M\,\{[\alpha]\ \pi_i\ \_ / [\alpha]\ \_\}$$
$$[\alpha]\ \mu\beta.M = M\,\{\alpha/\beta\} \qquad \mu\alpha.\,[\alpha]\ M = M \quad (\alpha \notin \mathrm{FV}(M)) \qquad \mu\alpha.M = M\,\{\ \_ / [\alpha]\ \_\} \quad (\alpha : 0)$$

$$\mathsf{succ}\,\overline{\mathsf{n}} = \overline{\mathsf{n}+\mathsf{1}} \qquad \mathsf{succ}\,(\mu\alpha.M) = \mu\alpha.M\,\{[\alpha]\ \mathsf{succ}\,(\_) / [\alpha]\ \_\}$$
$$\mathsf{rec}\,M\,N\,\overline{\mathsf{0}} = M \qquad \mathsf{rec}\,M\,N\,\overline{\mathsf{n}+\mathsf{1}} = N\overline{\mathsf{n}}\,(\mathsf{rec}\,M\,N\,\overline{\mathsf{n}}) \qquad \mathsf{rec}\,M\,N\,(\mu\alpha.P) = \mu\alpha.P\,\{[\alpha]\ \mathsf{rec}\,M\,N\,(\_) / [\alpha]\ \_\}$$

Figure 2. Equational theory of System μT

Scott domains and a category of continuations. The objects of $R^{\mathcal{C}}$ being pointed, we can interpret general fixpoints in $R^{\mathcal{C}}$, so in particular we can interpret bar recursion.

- The second example is the category of Hyland-Ong games [15] in its unbracketed version [18], which is known to interpret control operators. If $\mathcal{C}$ is the category of finite families of arenas and strategies, R is the singleton family of the one-move arena and $[\![I]\!]$ is the singleton family of the arena with countably many moves then $R^{\mathcal{C}}$ is a category of continuations which is isomorphic to the category of unbracketed games (see [6], sections 3.2.2 and 4.4.1). This category is cpo-enriched and can therefore also interpret bar recursion.

From now on we will drop the interpretation brackets for terms and use the syntax of $\lambda\mu$-calculus to manipulate morphisms of $R^{\mathcal{C}}$. We will allow weakening without mentioning it, that is, the injection from the homset $R^{\mathcal{C}}\,([T], [U] \parr [V])$ to the homset $R^{\mathcal{C}}\,([T] \times [T'], [U] \parr ([V] \parr [V']))$ obtained by precomposition with the left projection and postcomposition with $[U] \parr w^{l}_{[V],[V']}$ (with the notations of [22]) will be viewed as an inclusion:

$$R^{\mathcal{C}}\,([T], [U] \parr [V]) \subseteq R^{\mathcal{C}}\,\left([T] \times [T'], [U] \parr ([V] \parr [V'])\right)$$

3. Classical logic and the axiom of choice

In this section, we define our logical framework, which is based on classical logic with both strong and weak existentials (the latter being the double negations of the former). Formulas are divided into positive and negative, with classical reasoning being restricted to the negative ones. This is achieved by considering two-sided sequents in which at most one positive formula can appear on the right hand side, in principal position.

3.1 The proof system

Our logic is multi-sorted, with one base sort ι for natural numbers together with higher-order sorts built on it with the arrow constructor:

$$\sigma, \tau \quad ::= \quad \iota \mid \sigma \to \tau$$

There is an infinite set of first-order variables for each sort, and first-order terms are either variables, well-sorted applications or constants:

$$t^{\sigma}, u^{\tau} \quad ::= \quad x^{\sigma} \mid t^{\sigma \to \tau}\, u^{\sigma} \mid \mathsf{c}^{\sigma}$$

Since we are in a multi-sorted setting, we can have constants for combinators, natural numbers constructors, and recursor:

$$\mathsf{c}^{\sigma} \quad ::= \quad \mathsf{s}^{(\sigma\to\tau\to\nu)\to(\sigma\to\tau)\to\sigma\to\nu} \mid \mathsf{k}^{\sigma\to\tau\to\sigma} \mid 0^{\iota} \mid \mathsf{S}^{\iota\to\iota} \mid \mathsf{rec}^{\sigma\to(\iota\to\sigma\to\sigma)\to\iota\to\sigma}$$

We can write every term of Gödel's system T with these. In the following, we will often omit the sorting information of the first-order terms.

Since our logic encompasses both intuitionistic and classical proofs, we define both positive and negative formulas, by mutual induction. Full classical logic can be used on negative formulas, while the positive ones will be restricted to intuitionistic reasoning. Because we will perform a direct interpretation in System μT_{br} which has control features, the atomic formulas (inequalities between first-order terms of the same sort, and falsity) will be negative. Conjunctions of negative formulas, universal quantifications on negative formulas and implications with a negative conclusion are negative, while existentials of any formula are positive, as well as conjunctions with a positive component, implications with a positive conclusion and universal quantifications of positive formulas:

$$A^{-}, B^{-} \quad ::= \quad t^{\sigma} \neq_{\sigma} u^{\sigma} \mid \bot \mid A \Rightarrow B^{-} \mid A^{-} \wedge B^{-} \mid \forall x^{\sigma} A^{-}$$
$$A^{+}, B^{+} \quad ::= \quad A \Rightarrow B^{+} \mid A^{+} \wedge B \mid A \wedge B^{+} \mid \forall x^{\sigma} A^{+} \mid \exists x^{\sigma} A$$

Taking into account formulas on which classical reasoning is allowed as well as formulas which are restricted to intuitionistic reasoning relies on a distinction between positive and negative formulas introduced in [7] and related to proof systems LU [10], PCL [19] and LC [9] (see section 5.2).

Negation of a formula A is encoded as $\neg A \stackrel{\Delta}{=} A \Rightarrow \bot$ (hence it is always negative) and equality is encoded as $t^{\sigma} =_{\sigma} u^{\sigma} \stackrel{\Delta}{=} \neg (t^{\sigma} \neq_{\sigma} u^{\sigma})$. We will define our interpretation of proofs in System μT_{br} by directly annotating them with terms of System μT_{br}. In order to annotate the elimination rule of universal quantification and the introduction rule of existential quantification, we need to embed first-order terms into System μT_{br}. First, we interpret the

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \lambda p.p : x = x \mid \beta : \mathsf{S}\,0 = 0, \alpha : \exists x\,(x = 0)}}{\vdash \lambda xp.p : \forall x\,(x = x) \mid \beta : \mathsf{S}\,0 = 0, \alpha : \exists x\,(x = 0)}}{\vdash (\lambda xp.p)\,\overline{0} : 0 = 0 \mid \beta : \mathsf{S}\,0 = 0, \alpha : \exists x\,(x = 0)}}{\vdash \langle \overline{0}, (\lambda xp.p)\,\overline{0}\rangle : \exists x\,(x = 0) \mid \beta : \mathsf{S}\,0 = 0, \alpha : \exists x\,(x = 0)}}{\vdash [\alpha]\,\langle \overline{0}, (\lambda xp.p)\,\overline{0}\rangle : \bot \mid \beta : \mathsf{S}\,0 = 0, \alpha : \exists x\,(x = 0)}}{\vdash \mu\beta.\,[\alpha]\,\langle \overline{0}, (\lambda xp.p)\,\overline{0}\rangle : \mathsf{S}\,0 = 0 \mid \alpha : \exists x\,(x = 0)}}{\vdash \langle \overline{1}, \mu\beta.\,[\alpha]\,\langle \overline{0}, (\lambda xp.p)\,\overline{0}\rangle\rangle : \exists x\,(x = 0) \mid \alpha : \exists x\,(x = 0)}}{\vdash [\alpha]\,\langle \overline{1}, \mu\beta.\,[\alpha]\,\langle \overline{0}, (\lambda xp.p)\,\overline{0}\rangle\rangle : \bot \mid \alpha : \exists x\,(x = 0)}}{\vdash \mu\alpha.\,[\alpha]\,\langle \overline{1}, \mu\beta.\,[\alpha]\,\langle \overline{0}, (\lambda xp.p)\,\overline{0}\rangle\rangle : \exists x\,(x = 0) \mid}$$

Figure 3. An incorrect proof

sorts of the logic as types of System μT_{br}:

$$\iota^* \triangleq I \qquad (\sigma \to \tau)^* \triangleq \sigma^* \to \tau^*$$

Then, assuming that each first-order variable is a variable of System μT_{br}, we map each first-order term to a term of System μT_{br}:

$$x^* \triangleq x \qquad (t\,u)^* \triangleq t^* u^* \qquad \mathsf{s}^* \triangleq \lambda xyz.x\,z\,(y\,z)$$
$$\mathsf{k}^* \triangleq \lambda xy.x \qquad 0^* \triangleq \overline{0} \qquad \mathsf{S}^* \triangleq \mathsf{succ} \qquad \mathsf{rec}^* \triangleq \mathsf{rec}$$

Using this mapping, we define our proof system in which every derivation is annotated with a term of System μT_{br}. The sequents can have several formulas on the right, one of which is principal:

$$p_1 : A_1, \ldots, p_m : A_m \vdash M : C \mid \alpha_1 : {B_1}^-, \ldots, \alpha_n : {B_n}^-$$

The deduction rules are given in figure 4, and the set $\mathcal{A}x$ of axioms is given in figure 5. The axiom $(\exists x^\iota \neg A \Rightarrow \forall x^\iota A) \Rightarrow \neg\neg\forall x^\iota A$ (which is a variant of the usual double-negation shift) will allow us to derive the classical axiom of countable choice from its intuitionistic version in the next section. The meaning of the μ-variable κ of type I will be explained in section 4.

The handling of positive and negative formulas is simplified by our use of multi-conclusioned sequents with a principal formula on the right of each sequent. All formulas on the right-hand side of a sequent except the principal one must be negative, while the principal formula and the formulas on the left-hand side of a sequent can be either positive or negative. In particular, right contraction is forbidden on positive formulas, which corresponds to allowing only intuitionistic reasoning on these. This should be put in parallel with our use of $\lambda\mu$-calculus, in which there is a principal type on the right-hand side of a typing judgment, which is the type of the current term (the other types on the right-hand side are the types of the continuations variables). We give in figure 3 the implementation of the counter-example of [12] in our setting. An existential formula appears on the right-hand side in a non-principal position, and therefore this proof is invalid in our system. We will see at the end of section 4.2 that accepting such a proof in our system would lead to a degenerated realizability model.

3.2 Proof-theoretic strength

In this section, we discuss some properties of our hybrid proof system and relate it to more usual theories.

Non-extensional equality The equality in our system being defined by Leibniz's axiom schema, we easily have the following facts:

$$x^{\sigma\to\tau} =_{\sigma\to\tau} y^{\sigma\to\tau} \quad \Rightarrow \quad \forall z^\sigma\,(x^{\sigma\to\tau}\,z^\sigma =_\tau y^{\sigma\to\tau}\,z^\sigma)$$
$$x^\sigma =_\sigma y^\sigma \quad \Rightarrow \quad \forall z^{\sigma\to\tau}\,(z^{\sigma\to\tau}\,x^\sigma =_\tau z^{\sigma\to\tau}\,y^\sigma)$$

The following principle of extensionality:

$$\forall z^\sigma\,(x^{\sigma\to\tau}\,z^\sigma =_\tau y^{\sigma\to\tau}\,z^\sigma) \Rightarrow x^{\sigma\to\tau} =_{\sigma\to\tau} y^{\sigma\to\tau}$$

doesn't hold in general, since it is refuted in intensional models of system T, like game semantics. However, there exist extensional models of system T, like the category of Scott domains.

Primitive inequality We chose a primitive inequality rather than equality in order to be able to realize Leibniz's axiom schema. This technique introduced by Krivine is now common in classical realizability [17]. For the restriction of our system to negative formulas this does not change provability: a negative formula with equalities and/or inequalities is provable in our system (through the encoding $t = u \equiv \neg t \neq u$) if and only it is provable in the same system with negative equalities (through the encoding $t \neq u \equiv \neg t = u$). In particular, since in the classical fragment of our system (see the end of this section) every formula is negative, provability isn't changed for that fragment. However, for the full system and the intuitionistic subsystem, things are less clear and we leave this to future work.

Ex falso quodlibet The formula $\bot \Rightarrow A$ is provable in our system for any formula A. For negative formulas, we have the simple proof $\vdash \lambda p.\mu\alpha.p : \bot \Rightarrow A^- \mid$. For positive formulas, this proof is not correct because $\alpha : A^+$ is forbidden in our system. Nevertheless, there is still a proof of $\bot \Rightarrow A^+$, by induction on A: if we have $\vdash M : \bot \Rightarrow A \mid$ and $\vdash N : \bot \Rightarrow B^+ \mid$, then:

$$\vdash \lambda pq.N\,p : \bot \Rightarrow A \Rightarrow B^+ \mid \qquad \vdash \lambda p.\,\langle x, M\,p\rangle : \bot \Rightarrow \exists x A \mid$$
$$\vdash \lambda p.\,\langle N\,p, M\,p\rangle : \bot \Rightarrow B^+ \wedge A \mid \quad \vdash \lambda px.N\,p : \bot \Rightarrow \forall x B^+ \mid$$
$$\vdash \lambda p.\,\langle M\,p, N\,p\rangle : \bot \Rightarrow A \wedge B^+ \mid$$

Intuitionistic arithmetic with choice The connection with intuitionistic logic is rather direct, since taking the set of rules of figure 4 with sequents restricted to the form $\vec{p} : \vec{A} \vdash M : B \mid$ gives precisely intuitionistic logic. In particular, with that form of sequents, the rules for $\bot$ are removed, forbidding right contraction. Remark that in that case, $\bot \Rightarrow A$ is no longer provable so we have in fact only minimal logic. To get back full intuitionistic logic, we could have added $\bot \Rightarrow x \neq y$ as an axiom (which is realized in the model of section 4 by $\lambda p.p$), so $\bot \Rightarrow A$ would have an intuitionistic proof for any A, by induction on A. Considering its restriction to the intuitionistic fragment of the system, our realizability interpretation is similar to Kreisel's modified realizability such as presented in [5] for minimal logic, the main difference being that our interpretation is performed in the meta-theory rather than presented in the form of a logical translation.

When it comes to arithmetic, the set of axioms of figure 5 minus $(\exists x^\iota \neg A \Rightarrow \forall x^\iota A) \Rightarrow \neg\neg\forall x^\iota A$ is almost the set of axioms of intuitionistic arithmetic in higher types with choice. The only difference is that we have a primitive inequality, and our Leibniz's axiom schema concerns inequality. As stated above, the exact connection with intuitionistic arithmetic with a primitive equality is still to be investigated.

Classical arithmetic with countable choice Proofs of classical logic can be embedded in our system through a simple transformation of formulas: we transform every existential $\exists x A$ of classical logic into a weak existential $\exists^c x A \triangleq \neg\neg\exists x A$ in our system and work in the following restricted syntax of formulas:

$$A^c, B^c \quad ::= \quad t \neq u \mid \bot \mid A^c \Rightarrow B^c \mid A^c \wedge B^c \mid \forall x A^c \mid \exists^c x A^c$$

The negativity constraints on the right-hand side of our sequents are then automatically satisfied since A^c is always negative. Since only existential quantifications are modified by this transformation, classical logic can be embedded in our system if we prove that

$$\frac{}{\vec{p}:\vec{A},q:C\vdash q:C\mid\vec{\alpha}:\vec{B}^-}\qquad\frac{}{\vec{p}:\vec{A}\vdash M:C\mid\vec{\alpha}:\vec{B}^-}\ (M:C\in\mathcal{A}x)$$

$$\frac{\vec{p}:\vec{A},q:C\vdash M:D\mid\vec{\alpha}:\vec{B}^-}{\vec{p}:\vec{A}\vdash\lambda q.M:C\Rightarrow D\mid\vec{\alpha}:\vec{B}^-}\qquad\frac{\vec{p}:\vec{A}\vdash M:C\mid\vec{\alpha}:\vec{B}^-\qquad\vec{p}:\vec{A}\vdash N:D\mid\vec{\alpha}:\vec{B}^-}{\vec{p}:\vec{A}\vdash\langle M,N\rangle:C\wedge D\mid\vec{\alpha}:\vec{B}^-}\qquad\frac{\vec{p}:\vec{A}\vdash M:C^-\mid\beta:C^-,\vec{\alpha}:\vec{B}^-}{\vec{p}:\vec{A}\vdash[\beta]M:\bot\mid\beta:C^-,\vec{\alpha}:\vec{B}^-}$$

$$\frac{\vec{p}:\vec{A}\vdash M:C\Rightarrow D\mid\vec{\alpha}:\vec{B}^-\qquad\vec{p}:\vec{A}\vdash N:C\mid\vec{\alpha}:\vec{B}^-}{\vec{p}:\vec{A}\vdash MN:D\mid\vec{\alpha}:\vec{B}^-}\qquad\frac{\vec{p}:\vec{A}\vdash M:C_1\wedge C_2\mid\vec{\alpha}:\vec{B}^-}{\vec{p}:\vec{A}\vdash\pi_i\,M:C_i\mid\vec{\alpha}:\vec{B}^-}\qquad\frac{\vec{p}:\vec{A}\vdash M:\bot\mid\beta:C^-,\vec{\alpha}:\vec{B}^-}{\vec{p}:\vec{A}\vdash\mu\beta.M:C^-\mid\vec{\alpha}:\vec{B}^-}$$

$$\frac{\vec{p}:\vec{A}\vdash M:C\mid\vec{\alpha}:\vec{B}^-}{\vec{p}:\vec{A}\vdash\lambda x.M:\forall x\,C\mid\vec{\alpha}:\vec{B}^-}\ (x\notin\mathrm{FV}(\vec{A},\vec{B}^-))\qquad\frac{\vec{p}:\vec{A}\vdash M:\forall x\,C\mid\vec{\alpha}:\vec{B}^-}{\vec{p}:\vec{A}\vdash M\,t^*:C\,\{t/x\}\mid\vec{\alpha}:\vec{B}^-}$$

$$\frac{\vec{p}:\vec{A}\vdash M:A\,\{t/x\}\mid\vec{\alpha}:\vec{B}^-}{\vec{p}:\vec{A}\vdash\langle t^*,M\rangle:\exists x\,A\mid\vec{\alpha}:\vec{B}^-}\qquad\frac{\vec{p}:\vec{A},q:A\vdash N:C\mid\vec{\alpha}:\vec{B}^-\qquad\vec{p}:\vec{A}\vdash M:\exists x\,A\mid\vec{\alpha}:\vec{B}^-}{\vec{p}:\vec{A}\vdash(\lambda xq.N)\,(\pi_1\,M)\,(\pi_2\,M):C\mid\vec{\alpha}:\vec{B}^-}\ (M\notin\mathrm{FV}(\vec{A},C,\vec{B}^-))$$

Figure 4. Deduction rules of our proof system

$$\begin{array}{ll}
\lambda p.p:x=x & \lambda p.p:\neg A\Rightarrow A\,\{y/x\}\Rightarrow x\neq y\\
\lambda p.p:\mathsf{s}\,x\,y\,z=x\,z\,(y\,z) & [\kappa]\,\overline{0}:\mathsf{S}\,x\neq 0\\
\lambda p.p:\mathsf{k}\,x\,y=x & \mathsf{rec}:A\,\{0/x\}\Rightarrow\forall x^\iota\,(A\Rightarrow A\,\{\mathsf{S}\,x/x\})\Rightarrow\forall x^\iota\,A\\
\lambda p.p:\mathsf{rec}\,x\,y\,0=x & \lambda p.\,\langle\lambda x.\pi_1\,(p\,x)\,,\lambda x.\pi_2\,(p\,x)\rangle:\forall x^\sigma\exists y^\tau A\Rightarrow\exists v^{\sigma\to\tau}\forall x^\sigma A\,\{v\,x/y\}\\
\lambda p.p:\mathsf{rec}\,x\,y\,(\mathsf{S}\,z)=y\,z\,(\mathsf{rec}\,x\,y\,z) & \lambda pq.\mathsf{brec}\,p\,q\,\epsilon:(\exists x^\iota\neg A\Rightarrow\forall x^\iota A)\Rightarrow\neg\neg\forall x^\iota A
\end{array}$$

Figure 5. Axioms of our proof system

the usual rules of existential quantification are admissible for weak existentials $\exists^c$. For the introduction rule:

if $\quad\vec{p}:\vec{A}^c\vdash M:C^c\,\{t/x\}\mid\vec{\alpha}:\vec{B}^c$

then $\quad\vec{p}:\vec{A}^c\vdash\lambda q.q\,\langle t^*,M\rangle:\exists^c xC^c\mid\vec{\alpha}:\vec{B}^c$

and for the elimination rule:

if $\quad\vec{p}:\vec{A}^c,q:C^c\vdash M:D^c\mid\vec{\alpha}:\vec{B}^c$

and $\quad\vec{p}:\vec{A}^c\vdash N:\exists^c xC^c\mid\vec{\alpha}:\vec{B}^c$

then

$$\vec{p}:\vec{A}^c\vdash\mu\beta.N\,(\lambda r.\,[\beta]\,(\lambda xq.M)\,(\pi_1\,r)\,(\pi_2\,r)):D^c\mid\vec{\alpha}:\vec{B}^c$$

which is valid since D^c is negative.

The usual axioms of classical arithmetic are all satisfied (in particular, since every A^c is negative, provability is not changed by our use of a primitive inequality, see above). The last requirement is then that the axiom of countable choice $\forall x^\iota\exists^c y^\sigma A\Rightarrow\exists^c v^{\iota\to\sigma}\forall x^\iota A\,\{v\,x/y\}$ is provable in our system:

Lemma 1. *The following formula:*

$$\forall x^\iota\neg\neg\exists y^\sigma A\Rightarrow\neg\neg\exists v^{\iota\to\sigma}\forall x^\iota A\,\{v\,x/y\}$$

is provable in our system.

Proof. We will derive it from the two axioms:

$$(\exists x^\iota\neg B\Rightarrow\forall x^\iota B)\Rightarrow\neg\neg\forall x^\iota B$$
$$\forall x^\iota\exists y^\sigma A\Rightarrow\exists v^{\iota\to\sigma}\forall x^\iota A\,\{v\,x/y\}$$

As stated above $\bot\Rightarrow\forall x^\iota B$ is provable, therefore we can derive $\neg\exists x^\iota\neg B\Rightarrow\neg\neg\forall x^\iota B$ from the first axiom. We also have:

$$\vdash\lambda pq.p\,(\pi_1\,q)\,(\pi_2\,q):\forall x^\iota\neg\neg B\Rightarrow\neg\exists x^\iota\neg B\mid$$

so we get $\forall x^\iota\neg\neg B\Rightarrow\neg\neg\forall x^\iota B$, which is usually called double-negation shift. Next, we instantiate B with $\exists y^\sigma A$ to get:

$$\forall x^\iota\neg\neg\exists y^\sigma A\Rightarrow\neg\neg\forall x^\iota\exists y^\sigma A$$

Finally, the second axiom entails:

$$\neg\neg\forall x^\iota\exists y^\sigma A\Rightarrow\neg\neg\exists v^{\iota\to\sigma}\forall x^\iota A\,\{v\,x/y\}$$

so we get $\forall x^\iota\neg\neg\exists y^\sigma A\Rightarrow\neg\neg\exists v^{\iota\to\sigma}\forall x^\iota A\,\{v\,x/y\}$, which was our goal. □

Remark that the axiom $(\exists x^\iota\neg B\Rightarrow\forall x^\iota B)\Rightarrow\neg\neg\forall x^\iota B$ with B negative is actually derivable in our system, as is the double-negation shift for negative formulas. In the lemma, however, we instantiate it with $B\equiv\exists y^\sigma A$, which is a positive formula. Therefore we cannot use the classical proof, since control operators cannot be used on strong existentials, as was demonstrated in [12].

For simplicity reasons, we did not consider here the axiom of dependent choice. Realizing classical dependent choice in the current setting would be possible with the slightly different bar recursion operator of [7], but the version of the double-negation shift that we use here is not sufficient to derive classical dependent choice from intuitionistic choice because it allows shifting double negations over first-order quantifications of type ι only.

The embedding of classical proofs in our system that we just described is rough and doesn't exploit the existence of strong existentials. The purpose of our hybrid system being to mix strong and weak existentials in a single proof, we now explain how this is possible. For example, one lemma could state:

$$\exists u^{\iota\to\iota\to\iota}\exists v^{\iota\to\iota\to\iota}\forall x^\iota\forall y^\iota\begin{cases}y=(u\,x\,y)\times x+(v\,x\,y)\\\wedge\\v\,x\,y<x\end{cases}$$

with appropriate encodings of multiplication, addition and ordering using equality and System T. This lemma is clearly provable constructively, hence the strong existentials obtained using intuitionistic choice. Another lemma could be:

$$\exists^c u^{\iota\to\iota}\forall x^\iota\,(u\,x=0\Leftrightarrow\exists y_1\dots y_k P\,(y_1,\dots,y_k)=x)$$

for some polynomial P with coefficients in $\mathbb{N}$ and such that $\{P\,(n_1,\dots,n_k)\,|\,n_1,\dots,n_k\in\mathbb{N}\}$ is undecidable (such polynomial exists by Matiyasevich's negative answer to Hilbert's tenth problem). This lemma is provable using classical logic and the axiom of countable choice, which is why we only have a weak existential. Nevertheless, these two lemmas can be used in a single proof. When using the first lemma, the efficient realizer of intuitionistic choice will perform the computations, while when using the second, bar recursion will be used and unbounded search may be performed. In order to make weak and strong existentials interact, it may be necessary to turn a strong existential into a weak one, which is possible since we have $\vdash \lambda pq.q\,p : \exists x A\Rightarrow\exists^c x A\mid$. By doing that the computational efficiency is not lost, since we only encapsulate the strong existential into a weak one with possibilities of backtrack which are not used.

We now explain briefly why the interpretation of a strong existential is more efficient that the interpretation of a weak one. The first difference is on the types of the interpretation: while $(\exists x^\sigma A)^*=\sigma^*\times A^*$, the type interpretation of the corresponding weak existential is:

$$(\exists^c x^\sigma A)^*=((\exists x^\sigma A\Rightarrow\bot)\Rightarrow\bot)^*=(\sigma^*\times A^*\to 0)\to 0$$

The second difference concerns the computational behavior of the respective interpretations. On one hand, the interpretation of a strong existential consists of a witness (of type σ^*) and a proof (of type A^*) which are independent because of $R^{\mathcal{C}}$'s cartesianness. On the other hand, the interpretation of a weak existential takes an argument ϕ of type $\sigma^*\times A^*\to 0$ and can call it several times (this is called backtracking). In the general case, the arguments supplied to ϕ in a given call can even depend on the outcome of the previous calls. Therefore, using strong existentials whenever possible gives an interpretation that may be more efficient than the usual double-negation interpretation.

3.3 Soundness

We prove in this section that the System μT_{br} terms annotating the proofs are well-typed. We first map every formula A to a type A^* of System μT_{br}:

$$(t\neq u)^*\triangleq 0\qquad \bot^*\triangleq 0$$
$$(A\Rightarrow B)^*\triangleq A^*\to B^*\qquad (A\wedge B)^*\triangleq A^*\times B^*$$
$$(\forall x^\sigma A)^*\triangleq\sigma^*\to A^*\qquad (\exists x^\sigma A)^*\triangleq\sigma^*\times A^*$$

Then we prove that the term associated to a derivation has the type associated to the conclusion of the derivation. Note the presence of a μ-variable κ, which is there for extraction purposes. Its meaning will be explained in section 4.

Proposition 1 (Soundness)**.** *If a derivation has conclusion:*

$$\vec{p}:\vec{A}\vdash M:C\mid\vec{\alpha}:\vec{B}^-$$

and if FV $\left(\vec{A},C,\vec{B}^-\right)=\vec{x}^{\vec{\sigma}}$*, then there is a corresponding typing derivation of* M*:*

$$\vec{x}:\vec{\sigma}^*,\vec{p}:\vec{A}^*\vdash M:C^*\mid\vec{\alpha}:\vec{B}^{-*},\kappa:I$$

Proof. First, we prove that for any first-order term t^τ with FV $(t)=\vec{x}^{\vec{\sigma}}$ there is a typing derivation:

$$\vec{x}:\vec{\sigma}^*\vdash t^*:\tau^*\mid$$

by induction on t. Then the proof is by induction on the derivation. □

This result is typical of typed realizability, in which a realizer of a formula is a typed program, its type being inferred from the formula. Working in typed realizability allows in particular the use of models with strong properties, like categories of continuations in our case.

4. Realizability

We define here our realizability interpretation and prove that the System μT_{br} term annotating a proof of a formula is a realizer of that formula. Finally, we give two extraction results, for strong and weak existentials.

4.1 Realizability values

In this section, each formula gets an associated realizability value, which is a set of morphisms in $R^{\mathcal{C}}$. First, we set the range of the quantifiers by defining a set $|\sigma|\subseteq R^{\mathcal{C}}\,(\mathbf{1},[\sigma^*])$ for each sort σ:

$$|\iota|\triangleq\{\overline{\mathsf{n}}\,|\,n\in\mathbb{N}\}\qquad |\sigma\to\tau|\triangleq\{\phi\,|\,\forall\psi\in|\sigma|\,,\phi\,\psi\in|\tau|\}$$

For the sake of extraction, the realizability model is parameterized with a set $\perp\!\!\!\perp\subseteq|\iota|$, as in [7, 8] and similarly to [17]. This set is used to define a non-empty realizability value for the formula $\bot$, as usual in classical realizability. The realizability values are defined for closed formulas with parameters in $R^{\mathcal{C}}$, that is, formulas in which every free variable x^σ has been replaced with some element of $|\sigma|$. Again for the sake of extraction, the realizability value of a formula A will not be a subset of $R^{\mathcal{C}}\,(\mathbf{1},[A^*])$, but a subset of $R^{\mathcal{C}}\,(\mathbf{1},[A^*]\parr[I])$. Reasons for this choice are discussed before proving adequacy for the logical rules, and details can be found in [7]. Morphisms in this homset should be thought of as having a free μ-variable of type I, for which we use the reserved name κ. Intuitively, a realizer of A has two output channels: one is of type A^*, while the other is of type I. This corresponds to Krivine's distinction between realizers and proof-like realizers [17]. A realizer may output an element of $\perp\!\!\!\perp$ on the channel κ of type I, while a proof-like realizer must output on the principal channel of type A^*. Apart from the interpretation of the atomic formulas which takes into account the parameter $\perp\!\!\!\perp$, the interpretation of the other connectives is the same as in usual intuitionistic realizability like Kleene's or Kreisel's:

$$|\bot|\triangleq\{\phi\,|\,\mu\kappa.\phi\in\perp\!\!\!\perp\}$$
$$|t\neq u|\triangleq\begin{cases}|\bot| & \text{if } t^*=u^*\\ R^{\mathcal{C}}\,(\mathbf{1},[0]\parr[I]) & \text{otherwise}\end{cases}$$
$$|A\Rightarrow B|\triangleq\{\phi\,|\,\forall\psi\in|A|\,,\phi\,\psi\in|B|\}$$
$$|A\wedge B|\triangleq\{\phi\,|\,\pi_1\,\phi\in|A|\wedge\pi_2\,\phi\in|B|\}$$
$$|\forall x^\sigma A|\triangleq\{\phi\,|\,\forall\psi\in|\sigma|\,,\phi\,\psi\in|B\,\{\psi/x\}|\}$$
$$|\exists x^\sigma A|\triangleq\{\phi\,|\,\pi_1\,\phi\in|\sigma|\wedge\pi_2\,\phi\in|B\,\{\pi_1\,\phi/x\}|\}$$

Remember that we use implicit weakening, as stated at the end of section 2.2. This means that in the definition of $|\forall x^\sigma A|$, $\psi\in|\sigma|\subseteq R^{\mathcal{C}}\,(\mathbf{1},[\sigma^*])$ is considered as a morphism in $R^{\mathcal{C}}\,(\mathbf{1},[\sigma^*]\parr[I])$ when we apply ϕ to it, and in the definition of $|\exists x^\sigma A|$, while $\pi_1\,\phi$ is a morphism in $R^{\mathcal{C}}\,(\mathbf{1},[\sigma^*]\parr[I])$, $\pi_1\,\phi\in|\sigma|$ means that $\pi_1\,\phi$ is equal to some $\zeta\in|\sigma|\subseteq R^{\mathcal{C}}\,(\mathbf{1},[\sigma^*])$, when ζ is considered as a morphism in $R^{\mathcal{C}}\,(\mathbf{1},[\sigma^*]\parr[I])$.

4.2 Adequacy

In this section, we prove the adequacy of our realizability interpretation, that is, the interpretation in $R^{\mathcal{C}}$ of a System μT_{br} term annotating a proof of a formula is a realizer of that formula. First, we prove adequacy for the axioms, and then for the rules.

Adequacy for the axioms

• The equalities which are verified in $R^{\mathcal{C}}$ are realized by the identity:

Lemma 2. *If $\phi \in |\sigma|$ then $\lambda p.p \in |\phi = \phi|$.*

Proof. Since $|\phi = \phi| = |(\phi \neq \phi) \Rightarrow \bot|$ and $|\phi \neq \phi| = |\bot|$. □

Since $R^{\mathcal{C}}$ satisfies the equations of figure 2 and because of the interpretation of first-order terms in System μT_{br} given in section 3.1, the lemma above proves adequacy of the axiom of reflexivity and the definitional axioms of s, k and rec.

• Leibniz scheme is realized by the identity as well:

Lemma 3. *If $\phi, \psi \in |\sigma|$ then:*

$$\lambda p.p \in |\neg A\{\phi/x\} \Rightarrow A\{\psi/x\} \Rightarrow \phi \neq \psi|$$

Proof. If $\phi \neq \psi$ then $|\phi \neq \psi| = R^{\mathcal{C}}(\mathbf{1}, [0]\ \mathfrak{P}\ [I])$ and the result is trivial, and if $\phi = \psi$ then $|\phi \neq \psi| = |\bot|$ and $|A\{\phi/x\}| = |A\{\psi/x\}|$ so $|\neg A\{\phi/x\}| = |A\{\psi/x\} \Rightarrow \phi \neq \psi|$ □

• Anything realizes that $\mathsf{0}$ is no successor:

Lemma 4. *If $\overline{\mathsf{n}} \in |\iota|$ then $[\kappa]\,\overline{\mathsf{0}} \in |\mathsf{S}\,\overline{\mathsf{n}} \neq \mathsf{0}|$.*

Proof. $\overline{\mathsf{n+1}} \neq \overline{\mathsf{0}}$ by adequacy of $R^{\mathcal{C}}$ for System μT_{br} at type I, therefore we have $|\mathsf{S}\,\overline{\mathsf{n}} \neq \mathsf{0}| = R^{\mathcal{C}}(\mathbf{1}, [0]\ \mathfrak{P}\ [I])$ and the result is trivial. □

• Recursion realizes induction:

Lemma 5. $\mathsf{rec} \in |A\{\mathsf{0}/x\} \Rightarrow \forall x^{\iota}\,(A \Rightarrow A\{\mathsf{S}\,x/x\}) \Rightarrow \forall x^{\iota}\,A|$.

Proof. If $\phi \in |A\{\mathsf{0}/x\}|$ and $\psi \in |\forall x^{\iota}\,(A \Rightarrow A\{\mathsf{S}\,x/x\})|$, then $\mathsf{rec}\,\phi\,\psi\,\overline{\mathsf{n}} \in |A\{\overline{\mathsf{n}}/x\}|$ by induction on $n \in \mathbb{N}$. □

• The intuitionistic version of countable choice is trivially realized:

Lemma 6.

$$\lambda p.\,\langle \lambda x.\pi_1\,(p\,x)\,, \lambda x.\pi_2\,(p\,x)\rangle \in |\forall x^{\sigma}\exists y^{\tau}A \Rightarrow \exists v^{\sigma\to\tau}\forall x^{\sigma}A\{v\,x/y\}|$$

Proof. Immediate. □

• Adequacy of bar recursion for our version of the double-negation shift requires two assumptions on $R^{\mathcal{C}}$: sequence internalization and continuity. Sequence internalization means that any sequence of morphisms can be turned into a morphism of sequence type:

Definition 2 (Sequence internalization)**.** *If $(\phi_n)_{n\in\mathbb{N}}$ is a sequence of morphisms in $R^{\mathcal{C}}([T], [U]\ \mathfrak{P}\ [V])$, then there exists a morphism $\phi \in R^{\mathcal{C}}([T], [I \to U]\ \mathfrak{P}\ [V])$ such that for any $n \in \mathbb{N}$, $\phi\,\overline{\mathsf{n}} = \phi_n$.*

In particular, all functions on natural numbers must exist in the model, even the uncomputable ones. This is consistent with the fact that the combination of the axiom of choice with classical logic proves the existence of such functions. Sequence internalization is verified in the two examples given in section 2.2, Scott domains and game semantics, but it is of course not satisfied by the term model of System μT_{br}, and this is the main motivation for working in a model rather than directly with the syntactic language.

The second requirement is continuity:

Definition 3 (Continuity)**.** *If $\phi \in R^{\mathcal{C}}(\mathbf{1}, [(I \to T) \to 0]\ \mathfrak{P}\ [I])$, $\psi \in R^{\mathcal{C}}(\mathbf{1}, [I \to T]\ \mathfrak{P}\ [I])$ and $\mu\kappa.\phi\,\psi \in |\iota|$, then there exists $m \in \mathbb{N}$ such that for any $\zeta \in R^{\mathcal{C}}(\mathbf{1}, [I \to T]\ \mathfrak{P}\ [I])$:*

$$\left(\forall m' < m,\ \zeta\,\overline{\mathsf{m'}} = \psi\,\overline{\mathsf{m'}}\right) \Rightarrow \phi\,\zeta = \phi\,\psi$$

m is called the modulus of continuity of ϕ at ψ.

Continuity of the higher-order functional ϕ (which takes infinite objects as input) means that if the output of ϕ on input ψ is a natural number, then it only depends on a finite amount of ψ: the finite sequence $\psi\,\overline{\mathsf{0}}, \ldots, \psi\,\overline{\mathsf{m-1}}$ (where m is the modulus of continuity of ϕ at ψ). This requirement is also satisfied in Scott domains (it is an immediate consequence of Scott continuity) and in game semantics since in that case $R^{\mathcal{C}}$ 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 continuity, since we can for example consider a function which gives 0 if the input sequence is the constant 0 sequence, and 1 otherwise.

Using these two assumptions on $R^{\mathcal{C}}$, we can now prove adequacy of bar recursion for our version of the double-negation shift:

Lemma 7. $\lambda pq.\mathsf{brec}\,p\,q\,\epsilon \in |(\exists x^{\iota}\neg A \Rightarrow \forall x^{\iota}A) \Rightarrow \neg\neg\forall x^{\iota}A|$.

Proof. Let $\phi \in |\exists x^{\iota}\neg A \Rightarrow \forall x^{\iota}A|$ and $\psi \in |\neg\forall x^{\iota}A|$. The following fact is easily provable, using the equation defining brec:

$$\text{Let } \forall i < n, \zeta_i \in \left|A\left\{\overline{\mathsf{i}}/x\right\}\right| \text{ and write } \xi = \epsilon * \zeta_0 * \ldots * \zeta_{n-1}.$$
$$\text{If } \lambda x.\mathsf{brec}\,\phi\,\psi\,(\xi * x) \in |\neg A\{\overline{\mathsf{n}}/x\}| \text{ then } \mathsf{brec}\,\phi\,\psi\,\xi \in |\bot|$$

Now suppose for the sake of contradiction that $\mathsf{brec}\,\phi\,\psi\,\epsilon \notin |\bot|$. Then by the above fact we have:

$$\lambda x.\mathsf{brec}\,\phi\,\psi\,(\xi * x) \notin \left|\neg A\left\{\overline{\mathsf{0}}/x\right\}\right|$$

so there exists $\zeta_0 \in \left|A\left\{\overline{\mathsf{0}}/x\right\}\right|$ such that $\mathsf{brec}\,\phi\,\psi\,(\epsilon * \zeta_0) \notin |\bot|$. Iterating this argument gives us an infinite sequence $(\zeta_n)_{n\in\mathbb{N}}$ such that for every $n \in \mathbb{N}$, $\zeta_n \in |A\{\overline{\mathsf{n}}/x\}|$ and:

$$\mathsf{brec}\,\phi\,\psi\,(\epsilon * \zeta_0 * \ldots * \zeta_{n-1}) \notin |\bot|$$

We now internalize this sequence as $\varphi \in R^{\mathcal{C}}(\mathbf{1}, [I \to A^*]\ \mathfrak{P}\ [I])$ using our assumption on $R^{\mathcal{C}}$. But then $\varphi \in |\forall x^{\iota}A|$, so $\psi\,\varphi \in |\bot|$. Finally, using continuity of $R^{\mathcal{C}}$ (since $\mu\kappa.\psi\,\varphi \in \perp\!\!\!\perp \subseteq |\iota|$), let n be the modulus of continuity of ψ at φ and write $\xi = \epsilon * \zeta_0 * \ldots * \zeta_{n-1}$. We have:

$$\psi\,(\xi\ @\ \phi\,\langle\overline{\mathsf{n}}, \lambda x.\mathsf{brec}\,\phi\,\psi\,(\xi * x)\rangle) = \psi\,\varphi \in |\bot|$$

but this morphism is $\mathsf{brec}\,\phi\,\psi\,\xi$, which by construction is not in $|\bot|$, hence the contradiction. □

The proof above together with lemma 1 and adequacy for the logical rules (see next paragraph) proves that the computational interpretation of the axiom of countable choice for classical logic with bar recursion is adequate with respect to our realizability semantics. In [13], Herbelin defines a logic with strong existentials which are weakened to make them compatible with classical logic but which still allow the derivation of the axioms of countable and

dependent choices. While the classical axiom of dependent choice could be realized in our framework as well, Herbelin's approach in [13] seems rather different from ours. Herbelin restricts the elimination of strong existentials and uses an elaborate operational semantics with coinductive formulas to keep these compatible with classical logic. We choose to have both strong existentials (with an unrestricted elimination rule) on which no classical reasoning is allowed but which have a very easy computational interpretation, and weak existentials on which full classical logic can be used, but for which the axiom of countable choice requires the use of bar recursion. We believe that a detailed proof of normalization for the calculus of [13] would shed some lights on its connections with bar recursion, which would in turn clarify the relationship between his logic and the present work.

Adequacy for the logical rules Classical computations often rely on a duality between a program and its environment. In Krivine's classical realizability [17], a program is an element in a set Λ of terms, while an environment is an element in a set Π of stacks. Each formula gets both a realizability value which is a subset of Λ and a falsity value which is a subset of Π. Krivine's model is parameterized by an orthogonality relation $\perp\!\!\!\perp \subseteq \Lambda \times \Pi$ and each formula gets first a falsity value in $\mathcal{P}(\Pi)$ and then a realizability value in $\mathcal{P}(\Lambda)$ which is the orthogonal of its falsity value.

In our setting, programs are morphisms of the category $R^{\mathcal{C}}$, and environments are morphisms of the category $\mathcal{C}$. Since our realizability is typed, we do not have a set Λ of programs as in Krivine's untyped setting, but we have a set $R^{\mathcal{C}}(\mathbf{1}, [T])$ of programs of type T for each type T. The idea is then to define a typed orthogonality relation between $R^{\mathcal{C}}(\mathbf{1}, [T])$ and $\mathcal{C}(\mathbf{1}, [\![T]\!])$. In order to do that, we can simply perform evaluation (since $[T] = R^{[\![T]\!]}$) to get a morphism from $\mathbf{1}$ to R. This duality is similar to the one encountered in vector spaces, where the dual of some R-vector space E is the vector space E^* of linear forms from E to R. In this setting, an orthogonality relation is usually defined between E and E^* by stating that $x \in E$ and $\varphi \in E^*$ are orthogonal to each other if and only if $\varphi(x) = 0$ in R.

In order to define an orthogonality relation between $R^{\mathcal{C}}$ and $\mathcal{C}$, there should be at least two morphisms from $\mathbf{1}$ to R, but there is no reason for this to hold. For example, if $R^{\mathcal{C}}$ is the category of unbracketed Hyland-Ong games described in section 2.2, then R is the one-move arena and there is only one morphism from $\mathbf{1}$ to R. In order to extract programs computing natural numbers, we even need to have at least countably many such morphisms. To circumvent this issue, we will not define an orthogonality relation between $R^{\mathcal{C}}(\mathbf{1}, [T])$ and $\mathcal{C}(\mathbf{1}, [\![T]\!])$ but rather between $R^{\mathcal{C}}(\mathbf{1}, [T] \parr [I])$ and $\mathcal{C}([\![I]\!], [\![T]\!])$, based on our parameter $\perp\!\!\!\perp \subseteq |\iota|$. In a general way, we can combine morphisms:

$$\phi \in R^{\mathcal{C}}([U], [T] \parr [V]) \text{ and } ж \in \mathcal{C}([U] \times [\![V]\!], [\![T]\!])$$

by, first, partially uncurrying ϕ into $\Lambda^{-1}_{[\![V]\!]}\phi \in \mathcal{C}([U] \times [\![V]\!], [T])$, pairing it with ж, evaluating, and then currying the result to get:

$$[ж]\,\phi \in R^{\mathcal{C}}([U], [0] \parr [V])$$

The notation $[ж]\,\phi$ corresponds to the idea that ж is an environment that is substituted for the μ-variable $\alpha : T$ in $[\alpha]\,\phi$. In the particular case of $[U] = \mathbf{1}$ and $[V] = [I]$ we therefore get:

$$\begin{array}{l}\text{If } \phi \in R^{\mathcal{C}}(\mathbf{1}, [T] \parr [I]) \text{ and } ж \in \mathcal{C}([\![I]\!], [\![T]\!]) \\ \quad \text{then } [ж]\,\phi \in R^{\mathcal{C}}(\mathbf{1}, [0] \parr [I]) \text{ and } \mu\kappa.\,[ж]\,\phi \in R^{\mathcal{C}}(\mathbf{1}, [I])\end{array}$$

where κ is a reserved name for the free μ-variable of type I in $R^{\mathcal{C}}(\mathbf{1}, [T] \parr [I])$ and $R^{\mathcal{C}}(\mathbf{1}, [0] \parr [I])$ (see section 4.1). The orthogonality relation between $R^{\mathcal{C}}(\mathbf{1}, [T] \parr [I])$ and $\mathcal{C}([\![I]\!], [\![T]\!])$ can now be defined from our parameter $\perp\!\!\!\perp \subseteq |\iota|$:

Definition 4 (Orthogonality)**.** *For any type T we define an orthogonality relation between $R^{\mathcal{C}}(\mathbf{1}, [T] \parr [I])$ and $\mathcal{C}([\![I]\!], [\![T]\!])$ by:*

$$\phi \perp ж \quad \Longleftrightarrow \quad \mu\kappa.\,[ж]\,\phi \in \perp\!\!\!\perp$$

with $\phi \in R^{\mathcal{C}}(\mathbf{1}, [T] \parr [I])$ and $ж \in \mathcal{C}([\![I]\!], [\![T]\!])$.

That is the reason why in section 4.1 we chose to define the realizability value of a formula A as a set of morphisms in $R^{\mathcal{C}}(\mathbf{1}, [A^*] \parr [I])$ and we choose now to define the falsity value of A as a set of morphisms in $\mathcal{C}([\![I]\!], [\![A^*]\!])$. More details can be found in [7],

Until now there has been no semantic distinction between positive and negative formulas in our realizability interpretation because the distinction relates to the possibility of using classical reasoning or not and axioms are indifferent to the kind of logic used. The classical aspects of our proof system are entirely taken care of by the rules, which we now prove adequate. The distinction between positive and negative formulas is reflected in the realizability semantics by the fact that only the realizability value of a negative formula need to be the orthogonal of some falsity value. This is not necessary for positive formulas: orthogonality is only needed for classical logic. Therefore, only negative formulas A^- have a falsity value $\left\|A^-\right\| \subseteq \mathcal{C}([\![I]\!], [\![A^*]\!])$:

$$\|\bot\| \triangleq \mathcal{C}([\![I]\!], [\![0]\!]) \qquad \|t \neq u\| \triangleq \begin{cases} \|\bot\| & \text{if } t^* = u^* \\ \emptyset & \text{otherwise} \end{cases}$$

$$\left\|A \Rightarrow B^-\right\| \triangleq \left\{\mathbf{pair}\left(\Lambda^{-1}_{[\![I]\!]}\phi, ж\right) \;\middle|\; \phi \in |A| \wedge ж \in \|B\|\right\}$$

$$\left\|A^- \wedge B^-\right\| \triangleq \{\mathbf{inj}_1 \circ ж \mid ж \in \|A\|\} \cup \{\mathbf{inj}_2 \circ ж \mid ж \in \|B\|\}$$

$$\left\|\forall x^\sigma A^-\right\| \triangleq \left\{\mathbf{pair}\left(\Lambda^{-1}_{[\![I]\!]}\phi, ж\right) \;\middle|\; \phi \in |\sigma| \wedge ж \in \|A\{\phi/x\}\|\right\}$$

where $\Lambda^{-1}_{[\![I]\!]}$, $\mathbf{pair}(\_,\_)$, $\mathbf{inj}_i$ and $\circ$ are respectively partial uncurrying (from $R^{\mathcal{C}}(\mathbf{1}, [T] \parr [I])$ to $\mathcal{C}([\![I]\!], [T])$), pairing, injection and composition in $\mathcal{C}$.

It is now a property (rather than a definition as in Krivine's setting) that realizability values are the orthogonals of falsity values for negative formulas:

Lemma 8. *For any negative formula A^- with parameters:*

$$\phi \in \left|A^-\right| \quad \Longleftrightarrow \quad \forall ж \in \left\|A^-\right\|, \phi \perp ж$$

Proof. Follows from $\left[\mathbf{pair}\left(\Lambda^{-1}_{[\![I]\!]}(\psi), ж\right)\right]\phi = [ж]\,\phi\,\psi$ and $[\mathbf{inj}_i \circ ж]\,\phi = [ж]\,\pi_i\,\phi$ □

Since $|\exists x^\sigma A| = \bigcup_{\phi \in |\sigma|} \{\langle \phi, \psi\rangle \mid \psi \in |A\{\phi/x\}|\}$, the fact that $|\exists x^\sigma A|$ is not the orthogonal of some falsity value is also reminiscent of what happens in vector spaces: a union of linear subspaces is in general not a linear subspace, and therefore it is strictly included in its double orthogonal (which is the sum of the subspaces). Conversely, since we can also write $\left|\forall x^\sigma A^-\right| = \bigcap_{\phi \in |\sigma|} \left\{\psi \;\middle|\; \psi\,\phi \in \left|A^-\{\phi/x\}\right|\right\}$, this realizability value is stable by double orthogonality and has a corresponding falsity value, just like any intersection of linear subspaces is still a linear subspace. We can now state the adequacy lemma:

Proposition 2 (Adequacy)**.** *If a derivation has conclusion:*

$$\vec{p} : \vec{A} \vdash M : C \mid \vec{\alpha} : \vec{B}^-$$

and if $FV\left(\vec{A}, C, \vec{B}^-\right) \subseteq \vec{x}^{\vec{\sigma}}$, then for any $\vec{\phi} \in |\vec{\sigma}|$, and for any $\vec{\psi} \in \left|\vec{A}\left\{\vec{\phi}/\vec{x}\right\}\right|$ and $\vec{ж} \in \left\|\vec{B}^-\left\{\vec{\phi}/\vec{x}\right\}\right\|$, we have:

$$M\left\{\vec{\psi}/\vec{p}, \vec{ж}/\vec{\alpha}\right\} \in \left|C\left\{\vec{\phi}/\vec{x}\right\}\right|$$

Proof. First, we prove that for any first-order term t^τ with $\mathrm{FV}(t) \subseteq \vec{x}^{\vec{\sigma}}$ and for any $\vec{\phi} \in |\vec{\sigma}|$, we have $t^* \left\{\vec{\phi}/\vec{x}\right\} \in |\tau|$ by induction on t. Then the proof is by induction on the derivation and relies on the fact that the falsity values are only necessary for the negative formulas $\vec{B}^-$. □

We end this section by showing that if the proof of figure 3 was correct in our system, then the realizability model would be degenerated. This proves that the issue identified in [12] arises in our system as well. Suppose for the sake of contradiction that adequacy holds for the proof of figure 3. This means that $M \in |\exists x\,(x = \mathsf{0})|$, where:

$$M = \mu\alpha.\,[\alpha]\left\langle \overline{1}, \mu\beta.\,[\alpha]\left\langle \overline{0}, (\lambda x p.p)\,\overline{0}\right\rangle\right\rangle$$

Calculation shows that $\pi_1\, M = \overline{1}$ and $\pi_2\, M = \lambda p.p$, so we get $\lambda p.p \in \left|\overline{1} = \overline{0}\right|$, and since $\left|\overline{1} \neq \overline{0}\right| = R^{\mathcal{C}}\,(\mathbf{1}, [0]\,\mathfrak{R}\,[I])$, we obtain $|\bot| = R^{\mathcal{C}}\,(\mathbf{1}, [0]\,\mathfrak{R}\,[I])$ (remember that $\overline{1} = \overline{0}$ is $\left(\overline{1} \neq \overline{0}\right) \Rightarrow \bot$). This implies that $|t \neq u| = R^{\mathcal{C}}\,(\mathbf{1}, [0]\,\mathfrak{R}\,[I])$ regardless of wether $t^* = u^*$ or not, and therefore the realizability model is degenerated.

4.3 Extraction

Finally, we validate our model with two extraction results. Recall that $=_\sigma$ is the non-extensional equality defined in section 3.1 and discussed in section 3.2. The first extraction result is immediate and relies on strong existentials:

Proposition 3. *Suppose we have a derivation of some closed Π_2^0 formula:*

$$\vdash M : \forall x^\sigma \exists y^\tau\,(t^\nu =_\nu u^\nu)\ |$$

then for any $\phi \in |\sigma|$, we have $\pi_1\,(M\,\phi) \in |\tau|$ and moreover $t^*\,\{\phi/x, \pi_1\,(M\,\phi)\,/y\} = u^*\,\{\phi/x, \pi_1\,(M\,\phi)\,/y\}$ holds in $R^{\mathcal{C}}$.

Proof. We choose $\perp\!\!\!\perp = \emptyset$, therefore:

$$|t^*\,\{\phi/x, \pi_1\,(M\,\phi)\,/y\} = u^*\,\{\phi/x, \pi_1\,(M\,\phi)\,/y\}| \neq \emptyset$$
$$\Longleftrightarrow$$
$$t^*\,\{\phi/x, \pi_1\,(M\,\phi)\,/y\} = u^*\,\{\phi/x, \pi_1\,(M\,\phi)\,/y\} \text{ in } R^{\mathcal{C}}$$

And then the result is immediate by definition of the realizability values of strong existentials. We could also conclude by applying the intuitionistic axiom of choice to M. □

In particular, if we choose $\sigma = \tau = \iota$ and if we have a computationally adequate abstract machine to execute System μT_{br} (which is possible, as explained at the end of section 2.1), then for any $n \in \mathbb{N}$, $\pi_1\,(M\,\overline{\mathsf{n}})$ reduces to some $\overline{\mathsf{m}}$ such that $t^*\,\{\overline{\mathsf{n}}/x, \overline{\mathsf{m}}/y\}$ and $u^*\,\{\overline{\mathsf{n}}/x, \overline{\mathsf{m}}/y\}$ are equal in $R^{\mathcal{C}}$. Proposition 3 also holds for formulas of arbitrary complexity if we replace the concluding equality by the existence of a uniform realizer (a realizer with $\perp\!\!\!\perp = \emptyset$), but the goal here is to compare it to extraction from weak existentials, which holds only for Π_2^0 formulas.

The second extraction result concerns weak existentials:

Proposition 4. *Suppose we have a derivation of some closed formula of the form:*

$$\vdash M : \forall x^\sigma \neg\neg \exists y^\iota\,(t^\tau =_\tau u^\tau)\ |$$

then for any $\phi \in |\sigma|$, $\mu\kappa.M\,\phi\,(\lambda x.\pi_2\,x\,([\kappa]\,\pi_1\,x)) $ is some $\overline{\mathsf{n}} \in |\iota|$ such that $t^*\,\{\phi/x, \overline{\mathsf{n}}/y\} = u^*\,\{\phi/x, \overline{\mathsf{n}}/y\}$ holds in $R^{\mathcal{C}}$.

Proof. If $\phi \in |\sigma|$ then $M\,\phi \in |\neg\neg\exists y\,(t\,\{\phi/x\} = u\,\{\phi/x\})|$. We now fix $\perp\!\!\!\perp \subseteq |\iota|$ to be:

$$\perp\!\!\!\perp = \left\{\overline{\mathsf{m}} \,\middle|\, t^*\,\{\phi/x, \overline{\mathsf{m}}/y\} = u^*\,\{\phi/x, \overline{\mathsf{m}}/y\} \text{ holds in } R^{\mathcal{C}}\right\}$$

and we prove that:

$$\lambda x.\pi_2\,x\,([\kappa]\,\pi_1\,x) \in |\neg\exists y\,(t\,\{\phi/x\} = u\,\{\phi/x\})|$$

Let $\psi \in |\exists y\,(t\,\{\phi/x\} = u\,\{\phi/x\})|$. Then $\pi_1\,\psi \in |\iota|$ and $\pi_2\,\psi \in |\neg\,(t\,\{\phi/x, \pi_1\,\psi/y\} \neq u\,\{\phi/x, \pi_1\,\psi/y\})|$, so it suffices to prove that $[\kappa]\,\pi_1\,\psi \in |t\,\{\phi/x, \pi_1\,\psi/y\} \neq u\,\{\phi/x, \pi_1\,\psi/y\}|$. For that, we distinguish two cases:

- $\pi_1\,\psi \in \perp\!\!\!\perp$: in that case $\mu\kappa.\,[\kappa]\,\pi_1\,\psi = \pi_1\,\psi \in \perp\!\!\!\perp$, therefore $[\kappa]\,\pi_1\,\psi \in |\bot| \subseteq |t\,\{\phi/x, \pi_1\,\psi/y\} \neq u\,\{\phi/x, \pi_1\,\psi/y\}|$
- $\pi_1\,\psi \notin \perp\!\!\!\perp$: then by our choice of the parameter $\perp\!\!\!\perp$ we have $t^*\,\{\phi/x, \pi_1\,\psi/y\} \neq u^*\,\{\phi/x, \pi_1\,\psi/y\}$, and therefore we get $|t\,\{\phi/x, \pi_1\,\psi/y\} \neq u\,\{\phi/x, \pi_1\,\psi/y\}| = R^{\mathcal{C}}\,(\mathbf{1}, [0]\,\mathfrak{R}\,[I])$

Finally we get $M\,\phi\,(\lambda x.\pi_2\,x\,([\kappa]\,\pi_1\,x)) \in |\bot|$, and therefore $\mu\kappa.M\,\phi\,(\lambda x.\pi_2\,x\,([\kappa]\,\pi_1\,x)) \in \perp\!\!\!\perp$, which achieves the proof. □

Again, if we choose $\sigma = \tau = \iota$, then for any $n \in \mathbb{N}$, $\mu\kappa.M\,\overline{\mathsf{n}}\,(\lambda x.\pi_2\,x\,([\kappa]\,\pi_1\,x))$ reduces in the abstract machine to some $\overline{\mathsf{m}}$ such that $t^*\,\{\overline{\mathsf{n}}/x, \overline{\mathsf{m}}/y\}$ and $u^*\,\{\overline{\mathsf{n}}/x, \overline{\mathsf{m}}/y\}$ are equal in $R^{\mathcal{C}}$.

A variant of proposition 3 in the case $\tau = \iota$ can also be obtained from proposition 4 since weak existentials are derivable from the strong ones: if M is as in proposition 3 with $\tau = \iota$, then for any $\phi \in |\sigma|$, $\mu\kappa.\pi_2\,(M\,\phi)\,([\kappa]\,\pi_1\,(M\,\phi))$ is some $\overline{\mathsf{n}} \in |\iota|$ such that $t^*\,\{\phi/x, \overline{\mathsf{n}}/y\} = u^*\,\{\phi/x, \overline{\mathsf{n}}/y\}$ holds in $R^{\mathcal{C}}$.

5. Conclusion

We defined a realizability framework allowing the use of both strong and weak existential quantifications. Any proof in classical arithmetic with the axiom of countable choice can be interpreted as a program and a concrete value can be extracted from a proof of a Π_2^0 formula in this theory. Moreover, if some care is taken to use strong existentials whenever possible, the extracted program is more efficient that the ones obtained with the usual interpretations.

5.1 Future works

In the current setting, it is up to the person writing the proof to determine whether at some point the use of a strong existential is possible. However, no theoretic reason forbids this to be checked automatically. With an automatic procedure for this, one could work in the general setting of classical logic with countable choice and still obtain extracted programs which are as efficient as possible.

A concrete implementation of our framework would lead to a quantitative analysis of the efficiency of the extracted programs. In particular, we could do some efficiency comparison while staying in our framework, since the usual interpretation (with weak sums everywhere) is implementable directly in our system.

The impact of choosing a primitive inequality rather than an equality in our system is still to be investigated. Indeed, while provability is unchanged for the subsystem with only negative formulas as well as for its fragment with only weak existential quantifications (that fragment being equivalent to classical arithmetic with countable choice from the point of view of provability), this is still unclear for the full system or its intuitionistic restriction (without the right-hand context).

5.2 Related works

Our polarities are the ones defined in [7], where we mentioned a possible connection with Girard's LU proof system [10]. We recently became aware of the work of Liang and Miller [19] and believe that there is a strong connection between our polarities and the ones of their PCL system, despite their claim that the polarities of PCL are different from the ones of LU. In particular our system seems to be related to an extension of LC [9] with an intuitionistic

implication. However, any universal quantification is negative in LC, while in our system it is negative if and only if the formula quantified on is negative.

On the computational side, Herbelin managed to define in [13] a logic in which strong existentials are weakened just enough to be compatible with both classical logic and the axiom of countable choice, at the cost of a complicated computational interpretation involving corecursion and a lazy call-by-name evaluation strategy. We believe that there are strong connections between the operational semantics of Herbelin's calculus and bar recursion. This possible connection could appear from a careful analysis of a detailed proof of termination of Herbelin's calculus.

Refinements of program extraction were also investigated in [5] (with an application in [4]) in the context of negative translations. In [5], the authors identify *definite* and *goal* formulas, which allows them to double-negate only what they call the *critical* atoms. Since we work directly with classical proofs interpreted using control operators, we do not have to take care of double-negations in front of atomic formulas expicitly. On the other hand, the existentials of [5] are defined as "$\neg\forall\neg$" and are therefore equivalent to our weak existentials. They also consider a contructive existential (equivalent to our strong existential) that they write $\exists^*$ and which serves as a formal account of Friedman's trick. For that reason, its only legal use is as $\exists^* yG$ in a proof of $\forall x \neg \forall y \neg G$. This is very different from our setting where strong existentials are part of the syntax of formulas and can be manipulated as any other formula, provided the negativity condition is verified.

Raffalli described in [21] an extraction technique relying on an interaction between a classical proof of a forall-exists formula and an intuitionistic proof of decidability for the corresponding subformula. Despite the use of second-order logic, untyped calculus and the absence of strong existentials, his work has some similarities with ours. In order to take into account the intuitionistic nature of the proof of decidability, formulas of second-order logic are built from both intuitionistic and classical second-order variables, and double-negation elimination is restricted to classical variables. A formula is then said classical if it ends with a classical propositional variable. This distinction between intuitionistic and classical formulas is similar to our distinction between negative and positive formulas. This similarity can also be observed in the semantic interpretation of [21] where classical variables get an interpretation which is the orthogonal of some set of stacks, while intuitionistic variables have a primitive interpretation as a set of terms. This is very similar to our framework, where negative formulas have both a realizability and a falsity value which are orthogonal to each other, while positive formulas only have a primitive realizability value.

Acknowledgments

I would like to thank the anonymous reviewers for their helpful comments and suggestions. I also thank Camille Paoletti for her preliminary comments leading to significant improvements in the paper's readability. This research has been supported by the UK Engineering and Physical Sciences Research Council grant EP/K037633/1. No new data were created during this study.

References

[1] R. Amadio and P.-L. Curien. *Domains and Lambda-Calculi*, volume 46 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 1998.

[2] S. Berardi, M. Bezem, and T. Coquand. On the Computational Content of the Axiom of Choice. *Journal of Symbolic Logic*, 63(2):600–622, 1998.

[3] U. Berger and P. 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.

[4] U. Berger, H. Schwichtenberg, and M. Seisenberger. The Warshall Algorithm and Dickson's Lemma: Two Examples of Realistic Program Extraction. *Journal of Automated Reasoning*, 26(2):205–221, 2001.

[5] U. Berger, W. Buchholz, and H. Schwichtenberg. Refined program extraction from classical proofs. *Annals of Pure and Applied Logic*, 114(1-3):3–25, 2002.

[6] V. Blot. *Game semantics and realizability for classical logic*. PhD thesis, École Normale Supérieure de Lyon, 2014.

[7] V. Blot. Typed realizability for first-order classical analysis. *Logical Methods in Computer Science*, 11(4), 2015.

[8] V. Blot and C. 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.

[9] J. Girard. A New Constructive Logic: Classical Logic. *Mathematical Structures in Computer Science*, 1(3):255–296, 1991.

[10] J. Girard. On the Unity of Logic. *Annals of Pure and Applied Logic*, 59(3):201–217, 1993.

[11] T. Griffin. A Formulae-as-Types Notion of Control. In *17th Symposium on Principles of Programming Languages*, pages 47–58. ACM Press, 1990.

[12] H. Herbelin. On the Degeneracy of Sigma-Types in Presence of Computational Classical Logic. In *7th International Conference on Typed Lambda Calculi and Applications*, Lecture Notes in Mathematics, pages 209–220. Springer, 2005.

[13] H. Herbelin. A Constructive Proof of Dependent Choice, Compatible with Classical Logic. In *27th IEEE Symposium on Logic in Computer Science*, pages 365–374. IEEE Computer Society, 2012.

[14] M. Hofmann and T. Streicher. Completeness of Continuation Models for $\lambda\mu$-Calculus. *Information and Computation*, 179(2):332–355, 2002.

[15] M. Hyland and L. Ong. On Full Abstraction for PCF: I, II, and III. *Information and Computation*, 163(2):285–408, 2000.

[16] S. C. Kleene. On the Interpretation of Intuitionistic Number Theory. *Journal of Symbolic Logic*, 10(4):109–124, 1945.

[17] J.-L. Krivine. Realizability in classical logic. *Panoramas et synthèses*, 27:197–229, 2009.

[18] J. 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.

[19] C. Liang and D. Miller. Unifying Classical and Intuitionistic Logics for Computational Control. In *28th ACM/IEEE Symposium on Logic in Computer Science*, pages 283–292. IEEE Computer Society, 2013.

[20] M. 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.

[21] C. Raffalli. Getting results from programs extracted from classical proofs. *Theoretical Computer Science*, 323(1-3):49–70, 2004.

[22] P. Selinger. Control categories and duality: on the categorical semantics of the $\lambda\mu$ calculus. *Mathematical Structures in Computer Science*, 11(2):207–260, 2001.

[23] C. 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.