



**HAL**  
open science

# Realizability for Peano arithmetic with winning conditions in HON games

Valentin Blot

► **To cite this version:**

Valentin Blot. Realizability for Peano arithmetic with winning conditions in HON games. *Annals of Pure and Applied Logic*, 2017, 168 (2), pp.254 - 277. 10.1016/j.apal.2016.10.006 . hal-01766884

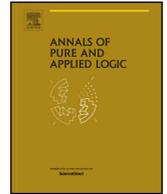
**HAL Id: hal-01766884**

**<https://hal.science/hal-01766884>**

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.



# Realizability for Peano arithmetic with winning conditions in HON games



Valentin Blot

*Department of Computer Science, University of Bath, United Kingdom*

## ARTICLE INFO

### Article history:

Available online 13 October 2016

### MSC:

03B70

03F30

68Q55

18D15

### Keywords:

Classical realizability

Peano arithmetic

Hyland Ong game semantics

## ABSTRACT

We build a realizability model for Peano arithmetic based on winning conditions for HON games. Our winning conditions are sets of desequentialized interactions which we call positions. We define a notion of winning strategies on arenas equipped with winning conditions. We prove that the interpretation of a classical proof of a formula is a winning strategy on the arena with winning condition corresponding to the formula. Finally we apply this to Peano arithmetic with relativized quantifications and give the example of witness extraction for  $\Pi_2^0$ -formulas.

© 2016 The Author. Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Realizability is a technique to extract computational content from formal proofs. It has been widely used to analyze intuitionistic systems (for e.g. higher-order arithmetic or set theory), see [23] for a survey. Following Griffin's computational interpretation of Peirce's law [8], Krivine developed in [13,14,12] a realizability for second-order classical arithmetic and Zermelo–Frænkel set theory.

On the other hand, Hyland–Ong game semantics provide precise models of various programming languages such as PCF [11] (a similar model has simultaneously been obtained in [21]), also augmented with control operators [15] and higher-order references [1]. In these games, plays are interaction traces between a program (player  $P$ ) and an environment (opponent  $O$ ). A program is interpreted by a strategy for  $P$  which represents the interactions it can have with any environment.

In this paper, we devise a notion of realizability for HON general games based on winning conditions on desequentialized plays, defined using the thick subtrees of [5]. We show that our model is sound for classical Peano arithmetic and allows to perform extraction for  $\Pi_2^0$ -formulas.

*E-mail address:* [v.blot@bath.ac.uk](mailto:v.blot@bath.ac.uk).

HON games with winning conditions on plays have been used in e.g. [6] for intuitionistic propositional logic with fixpoints. These conditions have then been generalized in [4] to handle full first-order classical logic, while [6] only deals with totality. In this paper we reformulate the winning conditions of [4] as sets of positions rather than sets of plays, emphasizing the fact that winning conditions are blind to the sequentiality information of plays. Classical logic is handled similarly to the unbracketed game model of PCF of [15].

We start from the cartesian closed category of single-threaded strategies which contains the unbracketed and non-innocent strategies used to model control operators and references. Applying the coproduct completion of [2] to this category gives a response category [22]. If we choose smartly the object of responses, the corresponding category of continuations is isomorphic to the first category and makes clear that the usual flat arena of natural numbers in HON games is indeed in the image of a negative translation. Our realizability is then obtained by equipping arenas with winning conditions on positions.

The paper is organized as follows. Section 2 introduces the notion of thick subtrees and positions, states the equivalence with the usual game semantics framework and shows how to interpret  $\lambda\mu$ -calculus in the category of games. Section 3 defines the notion of winning strategies. Section 4 contains the definition of our realizability relation and its adequacy for classical logic. Section 5 introduces the relativization predicate, applies our realizability model to Peano arithmetic and shows witness extraction for  $\Pi_2^0$ -formulas.

## 2. HON games

Our realizability model is based on the Hyland–Ong–Nickau games [11] with no bracketing or innocence constraint, so as to model control operators [15] and references [1]. We consider single-threaded strategies in order to have a cartesian closed category.

### 2.1. Arenas

Here we define arenas, which are forests of moves. Arenas are the objects of the category of HON games.

A forest is a partial order  $(E, \leq)$  such that  $\forall x \in E, \{y \in E \mid y < x\}$  is well-ordered. The binary relation  $<_1$  on  $E$  is then defined as:

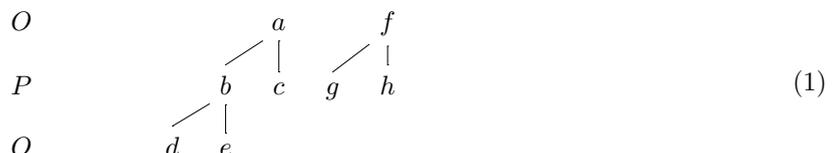
$$\forall x, y \in E \quad x <_1 y \iff x < y \wedge \forall z (x < z < y \Rightarrow z = x \vee z = y)$$

$x <_1 y$  means that  $y$  is a direct child of  $x$ . The roots of a forest correspond to the minimal elements for  $\leq$ .

**Definition 1 (Arena).** An arena is a countably branching, finite-depth forest whose nodes are called moves. Each move is given a polarity  $O$  (for Opponent) or  $P$  (for Player or Proponent):

- The roots are of polarity  $O$
- If  $x <_1 y$  then  $x$  and  $y$  are of opposite polarities

A root of an arena is also called an initial move. We will often identify an arena with its set of moves. Here is an example of arena, the polarities of the moves being given on the left:



2.2. Interaction sequences via thick subtrees

In usual game semantics [11], interaction sequences are defined as words of moves with pointers between them. Here we first define what is a position on an arena, using the thick subtrees of [5], and the interaction sequences are just positions together with a sequentiality information. This choice is justified by the positional nature of our winning conditions.

2.2.1. Interaction sequences as thick subtrees

We choose to define the positions on arenas using the thick subtrees of [5], extended to handle the case of forests. This formalism is a nice way to deal with non-affine programs: programs that may use several times their arguments. A thick subtree of a given tree is a subtree which can be extended in width, meaning that branches of the initial tree can be duplicated. These duplications correspond to the distinct computations of arguments during the execution of a non-affine program.

**Definition 2 (State).** Given an arena  $\mathcal{A}$ , a state on  $\mathcal{A}$  is a thick subforest of  $\mathcal{A}$ , that is a forest  $s$  together with a labeling function  $l : s \rightarrow \mathcal{A}$  such that:

$$\forall x \in s, \{l(y) \mid y < x\} = \{z \in \mathcal{A} \mid z < l(x)\}$$

This condition ensures that the roots are mapped to roots, and that the relation  $<_1$  is preserved. Here is an example of a state on the arena 1:

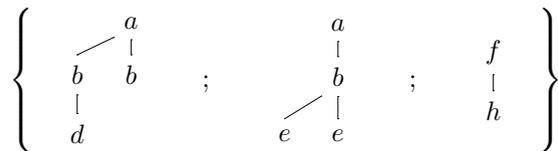


The nodes are considered distinct, even if they have the same label. By definition of the polarity, the roots of a justified sequence are labeled with  $O$ -moves, and if  $x <_1 y$ , then the labels of  $x$  and  $y$  are of opposite polarities.

We denote the empty state by  $\epsilon$ . If  $\mathcal{A}$  is an arena,  $X$  is a subset of  $\mathcal{A}$  (with the restricted ordering) and  $s$  is a state on  $\mathcal{A}$ , then  $s|_X$  is the state on  $X$  whose set of nodes is  $l^{-1}(X)$  (where  $l : s \rightarrow \mathcal{A}$  is the labeling of the state), with the restricted ordering. It is immediate to check that it is a state on  $X$ .

**Definition 3 (Position).** A position on an arena is a state which is a tree (i.e. it has a unique root). If  $s$  is a state on an arena  $\mathcal{A}$ , the set of positions of  $s$ ,  $\text{Pos}(s)$ , is the set of trees composing the state.

For example, the set of positions of the state 2 is:



A state  $s$  is a position if and only if  $\text{Pos}(s) = \{s\}$ . Remark that  $\text{Pos}(\epsilon) = \emptyset$ , so  $\epsilon$  is not a position. Our notion of position can be seen as an attempt to give a positional account of game semantics, as in [19], but in a quite different way.

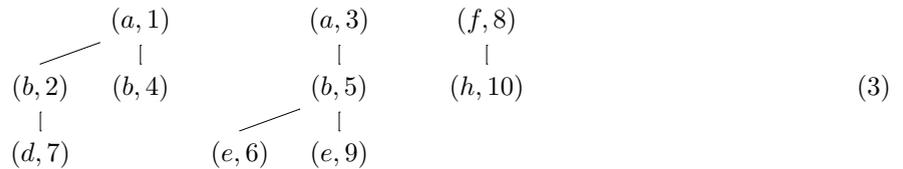
The justified sequences are then states on arenas together with a sequentiality information:

**Definition 4** (*Justified sequence*). Given an arena  $\mathcal{A}$ , we define a *justified sequence* on  $\mathcal{A}$  to be a state (finite or infinite)  $s$  on  $\mathcal{A}$  equipped with a total order  $\prec$  of type at most  $\omega$ .<sup>1</sup> Moreover, a justified sequence must verify:

$$\forall x, y \in s \quad x < y \quad \Rightarrow \quad x \prec y$$

The partial order  $<$  of a justified sequence corresponds to the pointers of usual game semantics settings:  $x <_1 y$  corresponds to  $y$  pointing to  $x$ . The total order  $\prec$  corresponds to the sequentiality of moves. The coherence condition of justified sequences means that if there is a pointer from a move  $y$  to a move  $x$  in the sequence, then  $x$  must have been played before  $y$ . The fact that  $x$  must be the father of  $y$  comes from the definition of a state.

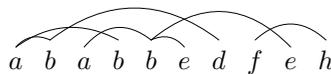
Here is an example of a justified sequence on the arena 1:



Where the labeling is given by the first components and the injection to natural numbers by the second components. The empty justified sequence will be denoted  $\epsilon$ , like the empty state. In the following we will allow ourselves to consider justified sequences as words of moves, without explicitly mentioning it. By the coherence condition of justified sequences, the minimal element for  $\prec$  is always a minimal element for  $<$ , and therefore labeled with a root of the arena. We can map any justified sequence to a state by forgetting the sequentiality information, so in the following we will freely consider that a justified sequence is a state. For example the state corresponding to the justified sequence 3 is the state 2.

2.2.2. Correspondence with the usual setting

In usual game semantics settings like [11], the justified sequences are words of moves with pointers between them. For example, the justified sequence 3 would be represented as:



We prove here the correspondence between our notion of justified sequence and the usual one of pointed sequence.

**Definition 5** (*Pointed sequence*). A (potentially infinite) pointed sequence on an arena  $\mathcal{A}$  is a word  $s$  of  $\mathcal{A}$  together with a partial pointing function  $f : |s| \rightarrow |s|$  such that:

- If  $f(i)$  is undefined then  $s_i$  is a root of  $\mathcal{A}$
- If  $f(i)$  is defined then  $f(i) < i$  and  $s_{f(i)} <_1 s_i$  (as elements of  $\mathcal{A}$ )

<sup>1</sup> This means that  $\prec$  can be described by an injection to natural numbers.

*From pointed sequences to justified sequences* If  $s$  is a pointed sequence with pointing function  $f$ , we define the corresponding justified sequence  $\text{Just}(s)$  as the thick subforest of  $\mathcal{A}$  whose nodes are  $(s_i, i)$ , the order  $<$  of the forest is defined as:

$$(s_i, i) < (s_j, j) \iff \exists n > 0, f^n(j) = i$$

the labeling function is given by the first components and the order  $<$  is given by the second components.

**Lemma 1.** *If  $s$  is a pointed sequence with pointing function  $f$ , then  $\text{Just}(s)$  is a state.*

**Proof.** Let  $(s_j, j)$  be a node. We have to prove:

$$\{s_i \mid (s_i, i) < (s_j, j)\} = \{x \in \mathcal{A} \mid x < s_j\}$$

For the left-to-right inclusion, if  $(s_i, i) < (s_j, j)$  then by definition there is some  $n > 0$  such that  $f^n(j) = i$ , so by the definition of a pointed sequence, in  $\mathcal{A}$  we have  $s_i = s_{f^n(j)} <_1 s_{f^{n-1}(j)} <_1 \dots <_1 s_{f(j)} <_1 s_j$ , and therefore  $s_i < s_j$ . For the right-to-left inclusion, since an arena has finite depth,  $\{x \in \mathcal{A} \mid x < s_j\}$  is finite, and it is also a well-order (by definition of a forest), so there are  $x_n <_1 \dots <_1 x_1 \in \mathcal{A}$  such that  $\{x \in \mathcal{A} \mid x < s_j\} = \{x_1; \dots; x_n\}$ . If  $x \in \mathcal{A}$  is such that  $x < s_j$ , then  $x = x_k$  for some  $1 \leq k \leq n$ , and then  $x = s_{f^k(j)}$ . Moreover, by definition  $(s_{f^k(j)}, f^k(j)) < (s_j, j)$  so we conclude.  $\square$

**Lemma 2.** *If  $s$  is a pointed sequence with pointing function  $f$ , then  $\text{Just}(s)$  is a justified sequence.*

**Proof.** We only have to prove that if  $(s_i, i) < (s_j, j)$ , then  $i < j$ . Indeed, if  $(s_i, i) < (s_j, j)$ , then by definition  $f^n(j) = i$  for some  $n > 0$ , but since  $f(k) < k$  as soon as  $f(k)$  is defined, we get immediately  $i < j$ .  $\square$

*From justified sequences to pointed sequences* If  $s$  is a justified sequence with labeling  $l$  we define here the corresponding pointed sequence  $\text{Point}(s)$ . We can order the elements of  $s$  using the total ordering  $\prec$ , which is of type at most  $\omega$ , and obtain a (possibly infinite) sequence  $l(x_0) \dots l(x_n) \dots$  such that  $x_i \prec x_j \iff i < j$ , which is the word of moves of  $\text{Point}(s)$ . We now define the pointing function  $f$  of  $\text{Point}(s)$ . If  $x_i \in s$ , there are two cases. Either  $\{x_j \in s \mid x_j < x_i\} = \emptyset$ , in which case  $f(i)$  is left undefined, either there is a unique  $x_j \in s$  such that  $x_j <_1 x_i$ , in which case we define  $f(i) = j$ .

**Lemma 3.** *If  $s$  is a justified sequence with labeling  $l$ , then  $\text{Point}(s)$  is a pointed sequence.*

**Proof.** Let  $f$  be the pointing function of  $\text{Point}(s)$ . If  $f(i)$  is undefined, then  $\{x_j \in s \mid x_j < x_i\} = \emptyset$ , but then:

$$\{x \in \mathcal{A} \mid x < l(x_i)\} = \{l(x_j) \mid x_j < x_i\} = \emptyset$$

so  $l(x_i)$  is a root of  $\mathcal{A}$ . If  $f(i) = j$ , then  $x_j <_1 x_i$  so  $x_j < x_i$  and  $x_j \prec x_i$  by definition of a justified sequence. By definition of the ordering that we chose, this means that  $j < i$ . Moreover, by definition of a state, since  $x_j <_1 x_i$  we get also  $l(x_j) <_1 l(x_i)$ .  $\square$

*Equivalence of the two notions* The two transformations above, from pointed sequences to justified sequences, and from justified sequences to pointed sequences, can be composed. We can prove that for any pointed sequence  $s$ ,  $\text{Point}(\text{Just}(s)) = s$ , and for any justified sequence  $s$ ,  $\text{Just}(\text{Point}(s)) = s$ . Justified sequences are in one-to-one correspondence with usual pointed sequences. In the following we will sometimes use “sequence” instead of “justified sequence”.

### 2.3. Strategies

**Definition 6 (Thread).** A thread on an arena is a sequence which, as a state, is a position. If  $s$  is a sequence on an arena  $\mathcal{A}$ , the set of threads of  $s$ ,  $\text{Threads}(s)$ , is the set of trees composing the sequence.

**Warning.** Note that a thread is a sequence which may not be alternating (see the definition of play), so our definition of thread differs from the usual one.

The threads are the analogue of positions but for sequences. A sequence  $s$  is a thread if it contains exactly one thread (i.e.  $\text{Threads}(s) = \{s\}$ ). Remark that  $\text{Threads}(\epsilon) = \emptyset$  and so  $\epsilon$  is not a thread. As for sequences, we will freely consider that a thread is a position.

**Definition 7 (Play).** A play  $s$  on  $\mathcal{A}$  is an alternating sequence on  $\mathcal{A}$ , i.e., the sequence of labels of the nodes ordered by  $\prec$  has alternating polarities.

A play on an arena is the trace of an interaction between a program and a context, each one performing an action alternatively. A  $P$ -play (resp.  $O$ -play) is a finite play whose maximal element for  $\prec$  is labeled with a  $P$ -move (resp.  $O$ -move), this means that as a word, it ends with a  $P$ -move (resp.  $O$ -move). If  $s$  and  $t$  are plays, write  $t \sqsubseteq s$  if  $t$  is an initial segment of  $s$  for  $\prec$  (with the order  $<$  restricted accordingly). This means that  $t$  is a prefix of  $s$  as a word, and their pointers coincide. Write  $t \sqsubseteq_P s$  (resp.  $t \sqsubseteq_O s$ ) if  $t$  is a  $P$ -prefix (resp.  $O$ -prefix) of  $s$ , i.e.  $t \sqsubseteq s$  and  $t$  is a  $P$ -play (resp.  $O$ -play). Write  $t \sqsubset_P s$  (resp.  $t \sqsubset_O s$ ) if  $t \sqsubseteq_P s$  (resp.  $t \sqsubseteq_O s$ ) and  $t \neq s$ .

**Definition 8 (Strategy).** A strategy  $\sigma$  on  $\mathcal{A}$  is a  $P$ -prefix-closed set of (finite)  $P$ -plays on  $\mathcal{A}$  such that:

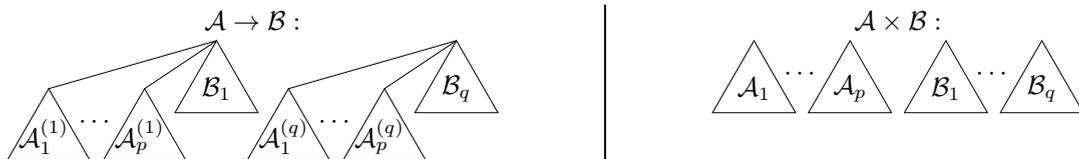
- $\sigma$  is deterministic: if  $sm$  and  $sm'$  are in  $\sigma$ , then  $m = m'$
- $\sigma$  is single-threaded: for any  $P$ -play  $s$ ,  $s \in \sigma \Leftrightarrow \text{Threads}(s) \subseteq \sigma$

Our notion of single-threadedness of strategies matches the usual one of thread-independence (see e.g. [1]). Also, a strategy always contains the empty play  $\epsilon$  since  $\text{Threads}(\epsilon) = \emptyset$ .

### 2.4. Cartesian closed structure

The constructions we use will sometimes contain multiple copies of the same arena (for example  $\mathcal{A} \rightarrow \mathcal{A}$ ), so we distinguish the instances with superscripts (for example  $\mathcal{A}^{(1)} \rightarrow \mathcal{A}^{(2)}$ ).

Let  $\mathcal{U}$  be the empty arena and  $\mathcal{V}$  be the arena with only one (opponent) move. If  $\mathcal{A}$  and  $\mathcal{B}$  are arenas consisting of the trees  $\mathcal{A}_1 \dots \mathcal{A}_p$  and  $\mathcal{B}_1 \dots \mathcal{B}_q$ , then the arenas  $\mathcal{A} \rightarrow \mathcal{B}$  and  $\mathcal{A} \times \mathcal{B}$  can be represented as follows:



The constructions described here define a cartesian closed category whose objects are arenas and morphisms are strategies. In the following this category will be denoted as  $\mathcal{C}$ . Details of the construction can be found in [9] and we only give a description of application and pairing of strategies:

**Lemma 4** (*Application*). Let  $\sigma$  be a strategy on arena  $\mathcal{A} \rightarrow \mathcal{B}$  and  $\tau$  be a strategy on arena  $\mathcal{A}$ . The strategy  $\sigma(\tau)$  on  $\mathcal{B}$  is:

$$\sigma(\tau) = \{s|_{\mathcal{B}} \mid s \in \sigma, s|_{\mathcal{A}} \in \tau \text{ and } s|_{\mathcal{B}} \text{ play on } \mathcal{B}\}$$

**Lemma 5** (*Pairing*). Let  $\sigma$  be a strategy on arena  $\mathcal{A}$  and  $\tau$  be a strategy on arena  $\mathcal{B}$ . The strategy  $\langle \sigma, \tau \rangle$  on  $\mathcal{A} \times \mathcal{B}$  is:

$$\langle \sigma, \tau \rangle = \{s \text{ play on } \mathcal{A} \times \mathcal{B} \mid s|_{\mathcal{A}} \in \sigma \text{ and } s|_{\mathcal{B}} \in \tau\}$$

These definitions of arenas will be used to associate arenas to the following simple types:

**Definition 9** (*Simple types*). The *simple types* are defined by the following grammar, where  $\iota$  is the unique base type:

$$T, U ::= \iota \mid \text{void} \mid \text{unit} \mid T \times U \mid T \rightarrow U$$

We suppose given an object  $\llbracket \iota \rrbracket$  of  $\mathcal{C}$  to interpret the base type, and we associate to each simple type  $T$  an object  $\llbracket T \rrbracket$  of  $\mathcal{C}$  as follows:

$$\llbracket \text{void} \rrbracket = \mathcal{V} \quad \llbracket \text{unit} \rrbracket = \mathcal{U} \quad \llbracket T \times U \rrbracket = \llbracket T \rrbracket \times \llbracket U \rrbracket \quad \llbracket T \rightarrow U \rrbracket = \llbracket U \rrbracket^{\llbracket T \rrbracket}$$

Since  $\mathcal{C}$  is cartesian closed, we can use the syntax of  $\lambda$ -calculus to define strategies from other strategies. In order to distinguish this notation from the  $\lambda\mu$ -terms of Sect. 2.6 we use a bold lambda  $\lambda$ . The type constant `void` is not essential for interpreting simply-typed  $\lambda$ -calculus in cartesian-closed  $\mathcal{C}$ , however it will become necessary for typing  $\lambda\mu$ -calculus in Sect. 2.6, and the interpretation of `void` as the one-move arena  $\mathcal{V}$  is essential for the continuation structure of  $\mathcal{C}$  presented in Sect. 2.5.

### 2.5. Category of continuations

We map classical proofs to strategies using the interpretation of call-by-name  $\lambda\mu$ -calculus in categories of continuations described in [22]. In order to see the category  $\mathcal{C}$  defined in Sect. 2.4 as a category of continuations, we use the Fam construction, a variant of the coproduct completion described in [2], so  $\mathcal{C}$  is isomorphic to  $R^{\text{Fam}(\mathcal{C})}$  for some well-chosen object  $R$  of  $\text{Fam}(\mathcal{C})$ . The Fam construction is defined as follows:

**Definition 10** ( $\text{Fam}(\mathcal{C})$ ). The objects of  $\text{Fam}(\mathcal{C})$  are families of objects of  $\mathcal{C}$  indexed by at most countable sets. A morphism from  $\{A_i \mid i \in I\}$  to  $\{B_j \mid j \in J\}$  is a function  $f : I \rightarrow J$  together with a family of morphisms of  $\mathcal{C}$  from  $A_i$  to  $B_{f(i)}$ , for  $i \in I$ .

Remark here that we differ from [2] because  $\mathcal{C}$  doesn't have weak coproducts nor all small products, and the families are countable. Thus  $\text{Fam}(\mathcal{C})$  is not bicartesian closed, but since  $\mathcal{C}$  is cartesian closed and has countable products,  $\text{Fam}(\mathcal{C})$  is still a distributive category with finite products and coproducts, and has exponentials of all singleton families. The empty product and terminal object is the singleton family  $\{\mathcal{U}\}$ , the empty sum and initial object is the empty family  $\{\}$ , and:

$$\{A_i \mid i \in I\} \times \{B_j \mid j \in J\} = \{A_i \times B_j \mid (i, j) \in I \times J\}$$

$$\{A_i \mid i \in I\} + \{B_j \mid j \in J\} = \{C_k \mid k \in I \uplus J\} \text{ where } C_k = \begin{cases} A_k & \text{if } k \in I \\ B_k & \text{if } k \in J \end{cases}$$

$$\{B_0\}^{\{A_i \mid i \in I\}} = \left\{ \prod_{i \in I} B_0^{A_i} \right\}$$

We fix once and for all:

$$R = \{\mathcal{V}\} = \{\llbracket \text{void} \rrbracket\}$$

which is an object of  $\text{Fam}(\mathcal{C})$  as a singleton family.  $R$  has all exponentials as stated above.  $\text{Fam}(\mathcal{C})$  is a response category and  $R^{\text{Fam}(\mathcal{C})}$ , the full subcategory of  $\text{Fam}(\mathcal{C})$  consisting of the objects of the form  $R^A$ , is a category of continuations (see [22]). The objects of  $R^{\text{Fam}(\mathcal{C})}$  are singleton families, and  $R^{\text{Fam}(\mathcal{C})}$  is isomorphic to  $\mathcal{C}$ . We will consider that objects and morphisms of  $R^{\text{Fam}(\mathcal{C})}$  are arenas and strategies and we will use the vocabulary defined at the end of Sect. 2.4 on  $R^{\text{Fam}(\mathcal{C})}$  also.

### 2.6. Interpreting the call-by-name $\lambda\mu$ -calculus

The types of  $\lambda\mu$ -calculus are the simple types of Definition 9. Let  $k^T$  range over a set of typed constants and  $x^T$  (resp.  $\alpha^T$ ) range over a countable set of variables (resp. names) for each type  $T$ . The grammar of  $\lambda\mu$ -terms is the following:

$$M, N ::= k^T \mid x^T \mid * \mid \langle M, N \rangle \mid \pi_1 M \mid \pi_2 M \mid \lambda x^T. M \mid M N \mid \mu \alpha^T. M \mid [\alpha^T] M$$

The typing rules can be found in [22], where our  $\text{unit}$  is their  $\top$ , our  $\times$  is their  $\wedge$  and our  $\text{void}$  is their  $\perp$ . For instance, the Law of Peirce is the type of the following closed term (we omit the type annotation of the variables).

$$\lambda x. \mu \alpha. [\alpha] s (\lambda y. \mu \beta. [\alpha] y) \text{ of type } ((T \rightarrow U) \rightarrow T) \rightarrow T \tag{4}$$

This  $\lambda\mu$ -term will be denoted  $\text{cc}$ .

We follow [22] to interpret call-by-name  $\lambda\mu$ -calculus in  $R^{\text{Fam}(\mathcal{C})}$ . In particular if  $M$  is a  $\lambda\mu$ -term of type  $T$  with free variables in  $x_1^{T_1}, \dots, x_n^{T_n}$  and no free name, then its interpretation is a morphism  $\llbracket M \rrbracket$  from  $\llbracket T_1 \rrbracket \times \dots \times \llbracket T_n \rrbracket$  to  $\llbracket T \rrbracket$ . This morphism coincides with the interpretation of the call-by-name CPS translation of  $M$  (defined in [22]) in the cartesian closed category  $R^{\text{Fam}(\mathcal{C})}$ . See [22] for details. As stated in [22], if two terms are equivalent under the call-by-name semantics of  $\lambda\mu$ -calculus, or equivalently if their call-by-name CPS translations are  $\beta\eta$ -equivalent, then their interpretations are the same.

In the following we will drop the double brackets for the interpretation of simple types.

### 3. Winning conditions on arenas

We will now define our notion of realizability. We equip arenas with winning conditions on positions. Realizers are then winning strategies, intuitively strategies whose positions are all winning.

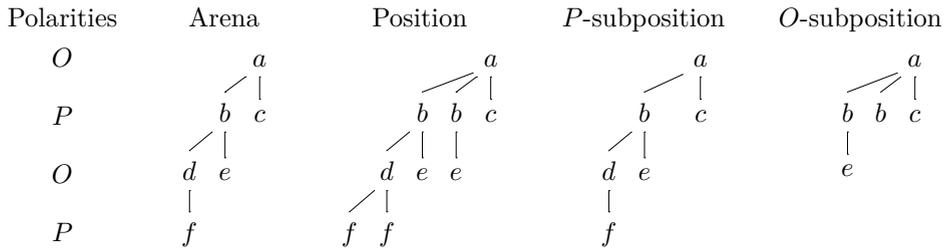
It is well-known that preservation of totality by composition of strategies is problematic in game semantics. Luckily we do not need to preserve totality, but only winningness. We thus do not impose any totality condition on strategies, but when it turns to the definition of winning positions, we have to take into account all maximal positions, including both infinite and odd-length ones. This leads to the notion of winning strategy proposed in Definition 15.

In order to define the notion of winning condition on an arena we introduce the notion of  $P$ -subposition and  $O$ -subposition:

**Definition 11** (*P-subposition, O-subposition*). If  $t$  is a position and  $u$  is a downward-closed subset of  $t$  (therefore it is a position), then  $u$  is a:

- $P$ -subposition of  $t$  if when  $n^P <_1 m^O$  in  $t$  and  $n^P \in u$ , then  $m^O \in u$
- $O$ -subposition of  $t$  if when  $n^O <_1 m^P$  in  $t$  and  $n^O \in u$ , then  $m^P \in u$

The intuitive meaning is that a  $P$ -subposition is a weakening of player (but any answer of opponent to a player move must be kept), and an  $O$ -subposition is a weakening of opponent (but any answer of player to an opponent move must be kept). A  $P$ -subposition (resp. an  $O$ -subposition) is obtained from a given position by cutting some subtrees whose roots are labeled with  $P$ -moves (resp.  $O$ -moves). Here is an example of a position together with a  $P$ -subposition and an  $O$ -subposition:



Now we can define the notion of winning condition on an arena:

**Definition 12** (*Winning condition*). A winning condition on  $\mathcal{A}$  is a set  $\mathcal{W}$  of positions on  $\mathcal{A}$  such that:

- If  $t$  is a position on  $\mathcal{A}$  and if some  $P$ -subposition of  $t$  is in  $\mathcal{W}$ , then  $t \in \mathcal{W}$ .
- If  $t \in \mathcal{W}$  then all the  $O$ -subpositions of  $t$  are in  $\mathcal{W}$ .

A winning position must be thought as a position that is winning for player. Under this interpretation, the two requirements read as: if some player-weakening of a position is already winning, then the full position is also winning, and if a position is winning, then any opponent-weakening of that position is all the more winning. The definition of the first stability constraint on winning conditions is motivated by our goal of interpreting classical logic. Indeed, Lemma 10 combined with this condition proves the validity of the strategy interpreting the `call/cc` operator with respect to the law of Peirce. The second condition is dual to the first and is necessary for the stability of winning conditions through the arrow construction, proved in Lemma 6.

A state  $s$  (and by extension a sequence or a play) on the arena  $\mathcal{A}$  equipped with the winning condition  $\mathcal{W}$  is said to be *winning* if  $\text{Pos}(s) \subseteq \mathcal{W}$ , which is consistent with the interpretation of “winning” as “winning for  $P$ ”. Our notion of winning state can be seen as a generalization of the ones defined in [6] and [10]. In order to obtain a realizability model of first-order logic, the notion of winning finite state is non-trivial and there can be odd-length plays which are winning and even-length plays which are losing. Winning conditions were also defined in [17] using payoffs on positions, but with different purposes, and in the framework of asynchronous games.

Remark that if  $t$  is a position on  $\mathcal{A} \rightarrow \mathcal{B}$ , then  $t|_{\mathcal{B}}$  is a position on  $\mathcal{B}$ , so  $t|_{\mathcal{B}}$  is winning iff  $t|_{\mathcal{B}} \in \mathcal{W}_{\mathcal{B}}$ , and if  $t$  is a position on  $\mathcal{A} \times \mathcal{B}$ , then  $t$  is either a position on  $\mathcal{A}$ , either a position on  $\mathcal{B}$ .

**Definition 13** (*Arrow and product of winning conditions*). If  $\mathcal{W}_{\mathcal{A}}$  and  $\mathcal{W}_{\mathcal{B}}$  are sets of positions on the arenas  $\mathcal{A}$  and  $\mathcal{B}$ , then we define:

$$\mathcal{W}_{\mathcal{A} \rightarrow \mathcal{B}} = \{t \text{ position on } \mathcal{A} \rightarrow \mathcal{B} \mid \text{Pos}(t|_{\mathcal{A}}) \subseteq \mathcal{W}_{\mathcal{A}} \Rightarrow t|_{\mathcal{B}} \in \mathcal{W}_{\mathcal{B}}\}$$

$$\mathcal{W}_{\mathcal{A} \times \mathcal{B}} = \left\{ t \text{ position on } \mathcal{A} \times \mathcal{B} \mid \begin{array}{l} t \text{ position on } \mathcal{A} \Rightarrow t \in \mathcal{W}_{\mathcal{A}} \\ t \text{ position on } \mathcal{B} \Rightarrow t \in \mathcal{W}_{\mathcal{B}} \end{array} \right\}$$

**Lemma 6.** If  $\mathcal{W}_{\mathcal{A}}$  and  $\mathcal{W}_{\mathcal{B}}$  are winning conditions on  $\mathcal{A}$  and  $\mathcal{B}$ , then  $\mathcal{W}_{\mathcal{A} \rightarrow \mathcal{B}}$  is a winning condition on  $\mathcal{A} \rightarrow \mathcal{B}$  and  $\mathcal{W}_{\mathcal{A} \times \mathcal{B}}$  is a winning condition on  $\mathcal{A} \times \mathcal{B}$ .

**Proof.**

- Let  $t$  be a position on  $\mathcal{A} \rightarrow \mathcal{B}$  and let  $u$  be a  $P$ -subposition of  $t$  such that  $u \in \mathcal{W}_{\mathcal{A} \rightarrow \mathcal{B}}$ . Suppose that  $\text{Pos}(t|_{\mathcal{A}}) \subseteq \mathcal{W}_{\mathcal{A}}$ . If  $v \in \text{Pos}(u|_{\mathcal{A}})$  then  $v$  is a  $O$ -subposition of some  $w \in \text{Pos}(t|_{\mathcal{A}}) \subseteq \mathcal{W}_{\mathcal{A}}$ , so  $v \in \mathcal{W}_{\mathcal{A}}$ . Then since  $u \in \mathcal{W}_{\mathcal{A} \rightarrow \mathcal{B}}$ ,  $u|_{\mathcal{B}} \in \mathcal{W}_{\mathcal{B}}$  and since  $u|_{\mathcal{B}}$  is a  $P$ -subposition of  $t|_{\mathcal{B}}$  we conclude that  $t|_{\mathcal{B}} \in \mathcal{W}_{\mathcal{B}}$ . Finally  $t \in \mathcal{W}_{\mathcal{A} \rightarrow \mathcal{B}}$ .
- Let  $t$  be a position on  $\mathcal{A} \rightarrow \mathcal{B}$  such that  $t \in \mathcal{W}_{\mathcal{A} \rightarrow \mathcal{B}}$  and let  $u$  be a  $O$ -subposition of  $t$ . Let suppose that  $\text{Pos}(u|_{\mathcal{A}}) \subseteq \mathcal{W}_{\mathcal{A}}$ . If  $v \in \text{Pos}(t|_{\mathcal{A}})$  and if  $m$  is the root of  $v$ , then the father of  $m$  is the root of  $t$  which is an  $O$ -move, and since  $u$  is a  $O$ -subthread of  $t$ , we get  $m \in u$ . Now the position of  $u|_{\mathcal{A}}$  whose root is  $m$  is in  $\mathcal{W}_{\mathcal{A}}$  and it is a  $P$ -subposition of  $v$ , so  $v \in \mathcal{W}_{\mathcal{A}}$ . Therefore  $\text{Pos}(t|_{\mathcal{A}}) \subseteq \mathcal{W}_{\mathcal{A}}$ , and since  $t \in \mathcal{W}_{\mathcal{A} \rightarrow \mathcal{B}}$  we have  $t|_{\mathcal{B}} \in \mathcal{W}_{\mathcal{B}}$ . Since  $u|_{\mathcal{B}}$  is a  $O$ -subposition of  $t|_{\mathcal{B}}$ , we get  $u|_{\mathcal{B}} \in \mathcal{W}_{\mathcal{B}}$ . Finally  $u \in \mathcal{W}_{\mathcal{A} \rightarrow \mathcal{B}}$ .
- Let  $t$  be a position on  $\mathcal{A} \times \mathcal{B}$ .  $t$  is either a position on  $\mathcal{A}$ , either a position on  $\mathcal{B}$ , so if  $u$  is a winning  $P$ -subposition of  $t$ , then either  $u \in \mathcal{W}_{\mathcal{A}}$ , either  $u \in \mathcal{W}_{\mathcal{B}}$ . Therefore  $t \in \mathcal{W}_{\mathcal{A}}$  or  $t \in \mathcal{W}_{\mathcal{B}}$ , and so  $t \in \mathcal{W}_{\mathcal{A} \times \mathcal{B}}$ .
- Let  $t$  be a position on  $\mathcal{A} \times \mathcal{B}$  such that  $t \in \mathcal{W}_{\mathcal{A} \times \mathcal{B}}$ . Either  $t \in \mathcal{W}_{\mathcal{A}}$ , either  $t \in \mathcal{W}_{\mathcal{B}}$ , so any  $O$ -subposition of  $t$  is in  $\mathcal{W}_{\mathcal{A}}$  or  $\mathcal{W}_{\mathcal{B}}$ , so in  $\mathcal{W}_{\mathcal{A} \times \mathcal{B}}$ .  $\square$

In order to define what a winning strategy is, we use a notion of augmented plays of a strategy inspired from [18]:

**Definition 14** (*Augmented play*). If  $\sigma$  is a strategy on  $\mathcal{A}$  and  $s$  is a play on  $\mathcal{A}$ , then  $s$  is an *augmented play* of  $\sigma$  if one of the following holds:

- $s \in \sigma$ , or
- $s$  is such that  $\forall t \sqsubset_P s, t \in \sigma$  and  $\forall t \in \sigma, s \not\sqsubseteq t$ .

In particular, in the second case of the above definition,  $s$  is either an  $O$ -play for which  $\sigma$  has no answer, either an infinite play (in which case  $s \sqsubseteq t \Leftrightarrow s = t$  and so the second condition, equivalent to  $s \notin \sigma$ , is always true since strategies contain only finite plays). Also, because of the second condition in the second case, an  $O$ -play  $s$  such that  $sm^P \in \sigma$  for some  $P$ -move  $m^P$  is not an augmented play of  $\sigma$ . Unlike [18], we consider not only odd-length extensions (with an  $O$ -move), but also infinite ones.

**Definition 15** (*Winning strategy*). If  $\sigma$  is a strategy on  $\mathcal{A}$  equipped with the winning condition  $\mathcal{W}$ , then  $\sigma$  is said to be *winning* if all its augmented plays are winning.

The following lemma will be useful to prove that a strategy  $\sigma$  is winning on  $(\mathcal{A}, \mathcal{W})$ .

**Lemma 7.** *If  $\sigma$  is a strategy on  $\mathcal{A}$  and if  $s$  is an augmented play of  $\sigma$ , then every  $t \in \text{Threads}(s)$  is an augmented play of  $\sigma$*

**Proof.**

- If  $s \in \sigma$ , then by single-threadedness of  $\sigma$ ,  $\text{Threads}(s) \subseteq \sigma$ .
- If  $s$  is an  $O$ -play, then we write  $s = s'm$  with  $s' \in \sigma$ . Let  $t \in \text{Threads}(s)$ . If  $m$  is not a move in  $t$ , then  $t \in \text{Threads}(s') \subseteq \sigma$ . If  $m$  is a move in  $t$ , then we write  $t = t'm$ , so  $t' \in \text{Threads}(s') \subseteq \sigma$ . If there is some  $n$  such that  $tn = t'mn \in \sigma$ , then  $\text{Threads}(s'mn) = (\text{Threads}(s') \setminus \{t'\}) \cup \{t'mn\} \subseteq \sigma$ , so by single-threadedness of  $\sigma$ ,  $sn = s'mn \in \sigma$ , contradicting the fact that  $s$  is an augmented play of  $\sigma$ .

- If  $s$  is infinite, let  $t \in \text{Threads}(s)$ . If  $t$  is finite, then there is some  $s' \sqsubseteq_P s$  such that  $t \in \text{Threads}(s')$ , but  $s' \in \sigma$ , so by single-threadedness of  $\sigma$   $t \in \sigma$ . If  $t$  is infinite, then for all  $t' \sqsubseteq_P t$  there is some  $s' \sqsubseteq_P s$  such that  $t' \in \text{Threads}(s')$ , but  $s' \in \sigma$ , so by single-threadedness of  $\sigma$ ,  $t' \in \sigma$ .  $\square$

The above result is the exact analogue of the left-to-right direction of the single-threadedness condition in Definition 8, but for the set of augmented plays of a strategy. The other direction is also true, but won't be used. Using this lemma it is sufficient to prove that every augmented play of  $\sigma$  which is a thread (let us call it an augmented thread of  $\sigma$ ) is in  $\mathcal{W}$  in order to prove that  $\sigma$  is winning on  $(\mathcal{A}, \mathcal{W})$ . Indeed, if any augmented thread of  $\sigma$  is in  $\mathcal{W}$  and if  $s$  is an augmented play of  $\sigma$ , then for any  $t \in \text{Threads}(s)$ ,  $t$  is an extended thread of  $\sigma$  by the lemma, so  $t \in \mathcal{W}$ . Then  $\text{Threads}(s) \subseteq \mathcal{W}$  so  $s$  is winning.

We now prove that the winning conditions on the arrow and product are compatible with application and pairing of strategies.

**Lemma 8.** *If  $\sigma$  is a winning strategy on  $(\mathcal{A} \rightarrow \mathcal{B}, \mathcal{W}_{\mathcal{A} \rightarrow \mathcal{B}})$  and  $\tau$  is a winning strategy on  $(\mathcal{A}, \mathcal{W}_{\mathcal{A}})$ , then  $\sigma(\tau)$  is a winning strategy on  $(\mathcal{B}, \mathcal{W}_{\mathcal{B}})$ .*

**Proof.** Let  $t$  be an augmented thread of  $\sigma(\tau)$ . Using the description of application given in Lemma 4, there is some augmented play  $u$  of  $\sigma$  such that  $u|_{\mathcal{A}}$  is an augmented play of  $\tau$  and  $u|_{\mathcal{B}} = t$ . Since  $t$  is a thread,  $u$  is also a thread, so since  $\sigma$  is winning on  $\mathcal{A} \rightarrow \mathcal{B}$ ,  $u \in \mathcal{W}_{\mathcal{A} \rightarrow \mathcal{B}}$ .  $u|_{\mathcal{A}}$  is an augmented play of  $\tau$  which is winning on  $\mathcal{A}$ , so  $\text{Threads}(u|_{\mathcal{A}}) \subseteq \mathcal{W}_{\mathcal{A}}$ , and therefore  $u|_{\mathcal{B}} \in \mathcal{W}_{\mathcal{B}}$ . Finally,  $t = u|_{\mathcal{B}} \in \mathcal{W}_{\mathcal{B}}$ . Therefore  $\sigma(\tau)$  is winning.  $\square$

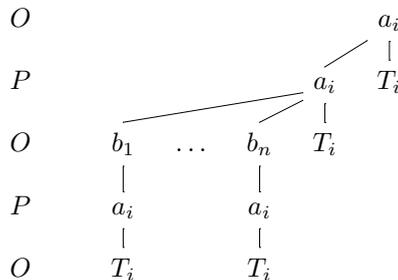
**Lemma 9.** *If  $\sigma$  is a winning strategy on  $(\mathcal{A}, \mathcal{W}_{\mathcal{A}})$  and  $\tau$  is a winning strategy on  $(\mathcal{B}, \mathcal{W}_{\mathcal{B}})$ , then  $\langle \sigma, \tau \rangle$  is a winning strategy on  $(\mathcal{A} \times \mathcal{B}, \mathcal{W}_{\mathcal{A} \times \mathcal{B}})$ .*

**Proof.** Let  $t$  be an augmented thread of  $\langle \sigma, \tau \rangle$ . Using the description of pairing given in Lemma 5 and the definition of the arena  $\mathcal{A} \times \mathcal{B}$ , and because  $t$  is a thread,  $t$  is either an augmented thread of  $\sigma$  on  $\mathcal{A}$ , in which case  $t \in \mathcal{W}_{\mathcal{A}}$ , either an augmented thread of  $\tau$  on  $\mathcal{B}$ , in which case  $t \in \mathcal{W}_{\mathcal{B}}$ . Therefore we get  $t \in \mathcal{W}_{\mathcal{A} \times \mathcal{B}}$ , and so  $\langle \sigma, \tau \rangle$  is winning.  $\square$

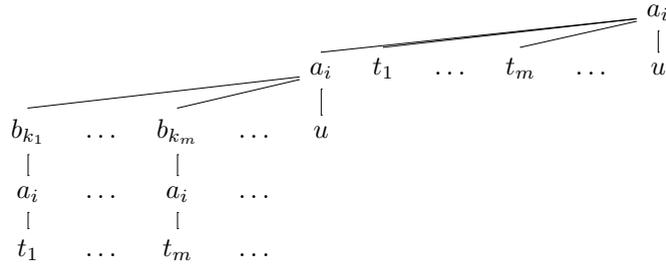
The following lemma on the interpretation of cc illustrates the use of winning conditions and justifies the notions of  $O$ - and  $P$ -subpositions in order to interpret classical logic through the law of Peirce.

**Lemma 10.** *If  $t$  is an augmented thread of  $\llbracket \text{cc} \rrbracket$  on the arena  $((T \rightarrow U) \rightarrow T) \rightarrow T$  (written  $((T^{(1)} \rightarrow U) \rightarrow T^{(2)}) \rightarrow T^{(3)})$ , then the positions of  $t|_{T^{(1)}}$  and  $t|_{T^{(2)}}$  are  $P$ -subpositions of  $t|_{T^{(3)}}$ .*

**Proof.**  $\llbracket \text{cc} \rrbracket$  being a total strategy, its augmented threads are even or infinite. Since  $t$  is a thread and player never plays in  $U$ ,  $t$  is a position on the subarena:



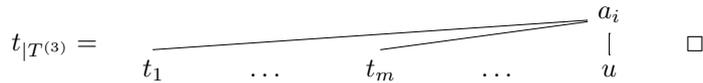
where  $T_i$  is a forest (every instance of  $T_i$  in the figure above is the same forest) such that  $\begin{matrix} a_i \\ | \\ T_i \end{matrix}$  is one of the trees of the arena  $T$ , and where  $b_1, \dots, b_n$  are the roots of the arena  $U$  (each  $b_i$  appearing zero or more times in  $t$ ). The position  $t$  is then of the following form:



where the  $t_j$  and  $u$  are positions on the arena  $T_i$ . With these notations we have:

$$\text{Pos}(t|_{T^{(1)}}) = \left\{ \begin{matrix} a_i \\ | \\ t_1 \end{matrix} ; \dots ; \begin{matrix} a_i \\ | \\ t_m \end{matrix} \right\} \text{ and } \text{Pos}(t|_{T^{(2)}}) = \left\{ \begin{matrix} a_i \\ | \\ u \end{matrix} \right\}$$

which are all  $P$ -subpositions of:



It follows easily from this lemma and Lemma 6 that for any winning conditions  $\mathcal{W}_T$  and  $\mathcal{W}_U$ ,  $\llbracket \text{cc} \rrbracket$  is winning on the arena

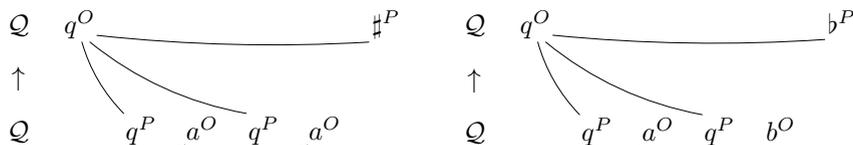
$$(((T \rightarrow U) \rightarrow T) \rightarrow T, \mathcal{W}_{((T \rightarrow U) \rightarrow T) \rightarrow T})$$

**Remark** ( $\mathcal{W}_{\mathcal{A} \rightarrow \mathcal{B}}$  versus Kleene arrow). Let  $\mathcal{A}, \mathcal{B}$  be arenas equipped with winning conditions  $\mathcal{W}_{\mathcal{A}}, \mathcal{W}_{\mathcal{B}}$ . We define here a strategy  $\sigma$  on  $\mathcal{A} \rightarrow \mathcal{B}$  such that for any winning strategy  $\tau$  on  $\mathcal{A}$ ,  $\sigma(\tau)$  is winning on  $\mathcal{B}$ , but  $\sigma$  is not winning on  $\mathcal{A} \rightarrow \mathcal{B}$ . Hence the arrow on winning conditions differs from the usual Kleene realizability arrow (see [23]).

We choose  $\mathcal{A}$  and  $\mathcal{B}$  to be the same arena  $\mathcal{Q}$  consisting of one root with three children  $\sharp, b$  and  $\natural$ , equipped with the winning condition

$$\mathcal{W}_{\mathcal{Q}} = \left\{ \begin{matrix} q^O \\ \swarrow \quad \downarrow \quad \searrow \\ a_1^P \quad a_2^P \quad \dots \end{matrix} \mid \exists i, a_i \in \{\sharp, \natural\} \right\}$$

where the positions may be finite or infinite. We define a strategy  $\sigma$  on  $\mathcal{Q} \rightarrow \mathcal{Q}$  such that for any  $\tau$  winning on  $(\mathcal{Q}, \mathcal{W}_{\mathcal{Q}})$ ,  $\sigma(\tau)$  is winning on  $(\mathcal{Q}, \mathcal{W}_{\mathcal{Q}})$ , but  $\sigma$  is not winning on  $(\mathcal{Q} \rightarrow \mathcal{Q}, \mathcal{W}_{\mathcal{Q} \rightarrow \mathcal{Q}})$ .  $\sigma$  is the innocent strategy defined by the views:



where  $a$  and  $b$  are distinct moves. The interaction with any single threaded strategy produces the left view, and so the projection  $q^O \#^P$  is winning, but the right view (which will never happen in an interaction with a single-threaded strategy) with  $a = \#$  and  $b = \natural$  is losing, so  $\sigma$  is losing.

#### 4. First-order logic

We define a realizability model for first-order classical logic with possibilities of witness extraction. For that the proposition  $\perp$  will be mapped to an arena in general different from  $\mathcal{V}$ . Its associated winning condition will be a parameter of the model, in the spirit of [14].

Let  $x$  range over a countable set of variables,  $f$  range over a set of function symbols with fixed finite arity and  $P$  range over a set of predicate symbols with fixed finite arity. First-order terms and formulas are defined by the following grammar:

$$\begin{aligned} a, b & ::= x \mid f(a_1, \dots, a_n) \\ A, B & ::= \perp \mid P(a_1, \dots, a_n) \mid \top \mid A \wedge B \mid A \Rightarrow B \mid \forall x A \end{aligned}$$

In the following we use syntactic sugar for the negation of formulas:  $\neg A \triangleq A \Rightarrow \perp$  and for the existential:  $\exists x A \triangleq \neg \forall x \neg A$ . Let  $\mathbf{Ax}$  be a set of closed formulas (the axioms of a theory). We use the following deduction system based on natural deduction with a rule for the law of Peirce. In these rules,  $\Gamma$  denotes a sequence of formulas  $A_1, \dots, A_n$ .

$$\begin{array}{c} \frac{}{\Gamma, A \vdash A} \quad \frac{}{\Gamma \vdash A} \quad (A \in \mathbf{Ax}) \quad \frac{}{\Gamma \vdash ((A \Rightarrow B) \Rightarrow A) \Rightarrow A} \\ \\ \frac{}{\Gamma \vdash \top} \quad \frac{\Gamma \vdash A \quad \Gamma \vdash B}{\Gamma \vdash A \wedge B} \quad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash A} \quad \frac{\Gamma \vdash A \wedge B}{\Gamma \vdash B} \\ \\ \frac{\Gamma, A \vdash B}{\Gamma \vdash A \Rightarrow B} \quad \frac{\Gamma \vdash A \Rightarrow B \quad \Gamma \vdash A}{\Gamma \vdash B} \\ \\ \frac{\Gamma \vdash A}{\Gamma \vdash \forall x A} \quad (x \notin \text{FV}(\Gamma)) \quad \frac{\Gamma \vdash \forall x A}{\Gamma \vdash A[a/x]} \end{array}$$

Remark that  $\perp$  has no associated rule, since the ex-falso rule has a particular status, given the interpretation of  $\perp$ . This will be discussed in Sect. 4.3.

We fix a countable first-order structure interpreting the terms of our logic, that is a countable set  $E$  together with an interpretation  $f^E : E^n \rightarrow E$  for each function symbol  $f$  of arity  $n$ . The interpretation is extended to every closed term: if  $a$  is a closed term of the logic, then  $a^E$  denotes its interpretation in the first-order structure, so  $a^E$  is an element of  $E$ . It will be convenient in the following to consider terms with parameters in  $E$ , which are first-order terms where some variables are substituted with elements of  $E$ . The interpretation extends naturally to such terms. In the same way we will also consider formulas with parameters: formulas in which first-order terms may have parameters.

##### 4.1. Realizability

We let  $\perp\!\!\!\perp$  be an arbitrary subset of  $E$ . We can map any first-order formula  $A$  to a simple type  $A^*$  as follows:

$$\begin{aligned} \perp^* &= \iota & (P(a_1, \dots, a_n))^* &= \iota & \top^* &= \text{unit} & (A \wedge B)^* &= A^* \times B^* \\ (A \Rightarrow B)^* &= A^* \rightarrow B^* & (\forall x A)^* &= A^* \end{aligned}$$

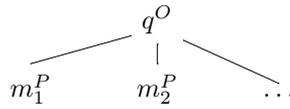
Remark that the type  $\perp^*$  is not the type `void` because the associated arena would be too small to hold informational content.

Recall that we omit the double bracket notation for the arenas, so a type  $T$  also denotes the associated arena. We fix the arena associated to  $\iota$  to be  $R^{(R^E)}$ , where  $\mathbf{E} = \{\mathcal{U}_e \mid e \in E\}$  is the countable family of empty arenas (and  $R = \{\mathcal{V}\}$ ). Hence  $\iota$  is the usual flat arena for the set  $E$ .

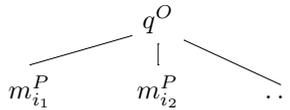
We define for any  $F \subseteq E$  the winning condition  $\mathcal{W}_F$  on  $\iota$  by:

$$\mathcal{W}_F = \left\{ \begin{array}{c} q^O \\ \swarrow \quad | \quad \searrow \\ m_1^P \quad m_2^P \quad \dots \end{array} \mid \exists i, m_i \in F \right\}$$

Note that it is a winning condition since the position:



that may be finite or infinite has only itself as  $O$ -subposition and



for  $1 \leq i_1 < i_2 \leq \dots$  as  $P$ -subpositions, and if  $\exists k, m_{i_k} \in F$  then  $\exists i, m_i \in F$ . Therefore  $\mathcal{W}_F$  is a winning condition on  $\iota$ . Let  $e \in E$ . The only extended thread of the usual strategy on  $\iota$  for  $e$  (the one that answers  $e$  to the first opponent move) is  $\begin{array}{c} q \\ | \\ e \end{array}$ , so this strategy is winning on  $\mathcal{W}_F$  as soon as  $e \in F$ .

We suppose given for each predicate  $P$  of arity  $n$  a set  $|P|$  of  $n$ -uples of elements of  $E$ . We can then define for each closed first-order formula  $A$  with parameters a winning condition  $\mathcal{W}_A$  on the arena  $A^*$ . We define  $\mathcal{W}_{A \wedge B} = \mathcal{W}_{A \times B}$  and  $\mathcal{W}_{A \Rightarrow B} = \mathcal{W}_{A \rightarrow B}$  as in [Definition 13](#),  $\mathcal{W}_\perp$  is  $\mathcal{W}_\perp$  (recall that  $\perp \subseteq E$ ), and we let:

$$\mathcal{W}_\top = \emptyset \quad \mathcal{W}_{\forall x A} = \bigcap_{e \in E} \mathcal{W}_{A[e/x]}$$

$$\mathcal{W}_{P(a_1, \dots, a_n)} = \begin{cases} \mathcal{W}_\perp & \text{if } (a_1^E, \dots, a_n^E) \notin |P| \\ \text{the set of all positions on } \iota & \text{otherwise} \end{cases}$$

Note that these are indeed winning conditions. For  $\mathcal{W}_\top$ , the empty set is a winning condition on  $\mathcal{U}$  which is the empty arena with no position. For  $\mathcal{W}_{\forall x A}$ , it is easy to see that an intersection of winning conditions is a winning condition, and for  $\mathcal{W}_{P(a_1, \dots, a_n)}$  and  $\mathcal{W}_\perp$  this is because  $\mathcal{W}_F$  is a winning condition for any  $F \subseteq E$ , and the set of all positions on  $\iota$  is trivially a winning condition on  $\iota$ .

Taking  $\mathcal{W}_\top = \emptyset$  may seem surprising at first, but remember that  $\top^* = \text{unit}$  is the empty arena with no threads ( $\epsilon$  is not a thread since  $\text{Threads}(\epsilon) = \emptyset$ ), so  $\mathcal{W}_\top = \emptyset$  is the only possibility. Nevertheless the only strategy on  $\mathcal{U}$ , which is  $\{\epsilon\}$ , is trivially winning since  $\text{Threads}(\epsilon) = \emptyset$ .

We can now define our notion of realizability:

**Definition 16 (Realizability relation).** If  $A$  is a closed first-order formula with parameters and if  $\sigma$  is a strategy on  $A^*$ , then  $\sigma$  realizes  $A$  (denoted  $\sigma \Vdash A$ ) if  $\sigma$  is a winning strategy on  $(A^*, \mathcal{W}_A)$ .

The following lemma shows that the identity formulas are realized by the corresponding identity strategies.

**Lemma 11.** *If  $A$  is a closed formula with parameters, then  $\llbracket \lambda x.x \rrbracket$  on  $A^* \rightarrow A^*$  is a realizer for the formula  $A \Rightarrow A$ .*

**Proof.** Let  $A^{*(1)} \rightarrow A^{*(2)}$  denote the arena  $A^* \rightarrow A^*$ . Let  $t$  be an augmented thread of  $\llbracket \lambda x.x \rrbracket$  such that  $\text{Threads}(t_{|_{A^{*(1)}}}) \subseteq \mathcal{W}_A$ . Since  $t$  is a thread,  $t_{|_{A^{*(2)}}}$  is a thread and  $t_{|_{A^{*(1)}}} = t_{|_{A^{*(2)}}$  is also a thread. Then  $t_{|_{A^{*(1)}}} \in \mathcal{W}_A$ , so  $t_{|_{A^{*(2)}}} = t_{|_{A^{*(1)}}} \in \mathcal{W}_A$ .  $\square$

The following result is a consequence of the remark following Lemma 10.

**Lemma 12.** *If  $A$  and  $B$  are closed formulas with parameters, then:*

$$\llbracket \text{cc} \rrbracket \Vdash ((A \Rightarrow B) \Rightarrow A) \Rightarrow A$$

#### 4.2. Adequacy for minimal classical logic

We now show that realizability is compatible with deduction in minimal classical logic. Full classical logic is discussed in Sect. 4.3.

##### 4.2.1. Translation of proofs to strategies

We use  $\lambda\mu$ -calculus and its interpretation in  $R^{\text{Fam}(\mathcal{C})} \cong \mathcal{C}$  to map a first-order proof to a typed  $\lambda\mu$ -term which is then interpreted in  $\mathcal{C}$  as a strategy.

Assume given a constant  $k_A$  of type  $A^*$  for each  $A \in \mathbf{Ax}$ . We map a derivation  $\nu$  of  $A_1, \dots, A_n \vdash A$  to a typed  $\lambda\mu$ -term  $\nu^*$  of type  $A^*$  with free variables in  $x^{A_1^*}, \dots, x^{A_n^*}$  as follows:

$$\begin{aligned} \frac{}{\Gamma, A \vdash A} &\rightsquigarrow x^{A^*} & \frac{}{\Gamma \vdash A} &^{(A \in \mathbf{Ax})} \rightsquigarrow k_A \\ \frac{}{\Gamma \vdash ((A \Rightarrow B) \Rightarrow A) \Rightarrow A} &\rightsquigarrow \text{cc} \text{ (see (4))} \\ \frac{}{\Gamma \vdash \top} &\rightsquigarrow * & \frac{\frac{\nu}{\Gamma \vdash A} \quad \frac{\nu'}{\Gamma \vdash B}}{\Gamma \vdash A \wedge B} &\rightsquigarrow \langle \nu^*, \nu'^* \rangle \\ \frac{\nu}{\Gamma \vdash A \wedge B} &\rightsquigarrow \pi_1 \nu^* & \frac{\nu}{\Gamma \vdash A \wedge B} &\rightsquigarrow \pi_2 \nu^* \\ \frac{\nu}{\Gamma, A \vdash B} &\rightsquigarrow \lambda x^{A^*}. \nu^* & \frac{\frac{\nu}{\Gamma \vdash A \Rightarrow B} \quad \frac{\nu'}{\Gamma \vdash A}}{\Gamma \vdash B} &\rightsquigarrow \nu^* \nu'^* \\ \frac{\nu}{\Gamma \vdash \forall x A} &_{(x \notin \text{FV}(\Gamma))} \rightsquigarrow \nu^* & \frac{\nu}{\Gamma \vdash \forall x A} &\rightsquigarrow \nu^* \end{aligned}$$

##### 4.2.2. Adequacy

We now prove that the strategies interpreting the proofs are realizers of the proved formula. If  $A$  is a formula and  $\theta$  an assignment of elements of  $E$  to variables, then  $A[\theta]$  is the formula with parameters obtained by replacing each first-order variable of  $A$  with its image by  $\theta$ .

**Lemma 13.** *Let  $\perp \subseteq E$ . Suppose that we have a realizer  $\llbracket k_A \rrbracket$  for each formula  $A \in \mathbf{Ax}$ . If  $\nu$  is a derivation of the sequent  $\Gamma \vdash A$  and if  $\theta$  is an assignment of elements of  $E$  to free variables of  $\Gamma, A$ , then  $\llbracket \nu^* \rrbracket$  is a winning strategy on  $\Gamma^* \rightarrow A^*$  equipped with  $\mathcal{W}_{\Gamma[\theta] \Rightarrow A[\theta]}$ .*

**Proof.** We prove the property by induction on the proof tree. For the axiom rules, we use in one case Lemma 11, and in the other case this is an assumption of the lemma. The adequacy for the law of Peirce comes from Lemma 10 and that of the introduction rule of  $\top$  is immediate since  $\text{Threads}(\llbracket * \rrbracket) = \emptyset$ . The introduction of  $\wedge$  follows from Lemma 9, the elimination of  $\Rightarrow$  from Lemma 8 and for its introduction the induction property doesn't change. The remaining rules are the elimination of  $\wedge$  and the introduction and elimination of  $\forall$ :

- For the first elimination rule of  $\wedge$ , let  $\nu$  be a proof of  $\Gamma \vdash A \wedge B$ . An augmented thread of  $\llbracket \pi_1 \nu^* \rrbracket$  is an augmented thread of  $\llbracket \nu^* \rrbracket$  which is a thread on  $\Gamma^* \rightarrow A^*$ , so by Definition 13 it is winning on  $\Gamma[\theta] \Rightarrow A[\theta] \wedge B[\theta]$
- The case of the second elimination rule of  $\wedge$  is similar
- For the introduction of  $\forall$ , let  $\nu$  be a proof of  $\Gamma \vdash A$  with  $x \notin \text{FV}(\Gamma)$ . If  $t$  is an augmented thread of  $\llbracket \nu^* \rrbracket$  such that:

$$\text{Threads}(t_{|\Gamma^*}) \subseteq \mathcal{W}_{\Gamma[\theta]}$$

Let  $e \in E$  and let  $\theta' = \theta \cup \{x \mapsto e\}$ . Since  $x \notin \text{FV}(\Gamma)$ ,  $\Gamma[\theta'] = \Gamma[\theta]$  so:

$$\text{Threads}(t_{|\Gamma^*}) \subseteq \mathcal{W}_{\Gamma[\theta']}$$

And by induction hypothesis  $t_{|A^*} \in \mathcal{W}_{A[\theta']}$ . Since  $A[\theta'] = A[e/x][\theta]$ , we get that for all  $e \in E$ ,  $t_{|A^*} \in \mathcal{W}_{A[e/x][\theta]}$ , and so  $t_{|A^*} \in \mathcal{W}_{(\forall x A)[\theta]}$

- For the elimination of  $\forall$ , let  $\nu$  be a proof of  $\Gamma \vdash \forall x A$  and let  $a$  be a first-order term. If  $t$  is an augmented thread of  $\llbracket \nu^* \rrbracket$  such that:

$$\text{Threads}(t_{|\Gamma^*}) \subseteq \mathcal{W}_{\Gamma[\theta]}$$

Then by induction hypothesis, since  $(a[\theta])^E \in E$  we get:

$$t_{|A^*} \in \mathcal{W}_{A[(a[\theta])^E/x][\theta]}$$

which terminates the proof since  $A \left[ (a[\theta])^E/x \right] [\theta] = (A[a/x])[\theta]$ .  $\square$

### 4.3. Full classical logic

In order to get full classical logic we need to add an ex-falso rule. However since the arena  $\perp^*$  is not the empty arena  $\mathcal{U}$  (see Sect. 4.1), we have to ensure that  $(\iota, \mathcal{W}_\perp)$  is included in  $(A^*, \mathcal{W}_A)$  for any formula  $A$ . This means that  $\iota$  is a subarena of  $A^*$ , so a position on  $\iota$  is in particular a position on  $A^*$ , and that  $\mathcal{W}_\perp \subseteq \mathcal{W}_A$ . This result is obtained for formulas not of the form  $A \Rightarrow \top \wedge \dots \wedge \top$  by an easy induction on formulas, since the winning conditions associated to atomic formulas different from  $\top$  are either  $\mathcal{W}_\perp$ , either every position on  $\iota$ . Formulas of the form  $A \Rightarrow \top \wedge \dots \wedge \top$  being trivially derivable, we can add to our deduction system the following rule:

$$\frac{\Gamma \vdash \perp}{\Gamma \vdash A}$$

where  $A$  is any first-order formula (defined in Sect. 4). This rule is erased by the mapping from derivations to  $\lambda\mu$ -terms (or replaced with a trivial derivation for formulas of the form  $A \Rightarrow \top \wedge \dots \wedge \top$ ). The corresponding adequacy lemma is immediate from Lemma 13.

#### 4.4. First-order logic with equality

We now show how to handle equality. We suppose that our first-order language contains an inequality predicate  $\neq$  of arity 2 for which we use infix notation. As stated in Sect. 4.1, this predicate is interpreted with the base type  $\iota$ . The set of pairs associated to  $\neq$  is:

$$|\neq| = \{(e, f) \in E^2 \mid e \neq f\}$$

In the following we use the notation  $(a = b) \triangleq \neg(a \neq b)$ . The following lemma states that the equations which are satisfied in the model are trivially realized:

**Lemma 14.** *Let  $a, b$  be first-order terms with free variables  $x_1, \dots, x_n$ . If:*

$$\forall e_1, \dots, e_n \in E, (a[e_1/x_1, \dots, e_n/x_n])^E = (b[e_1/x_1, \dots, e_n/x_n])^E$$

then:

$$\llbracket \lambda x.x \rrbracket \Vdash \forall x_1 \dots \forall x_n a = b$$

**Proof.** We write  $\vec{x}$  for  $x_1 \dots x_n$  and  $\vec{e}$  for  $e_1 \dots e_n$ .  $\llbracket \lambda x.x \rrbracket$  is a strategy on the arena:

$$\iota^{(1)} \rightarrow \iota^{(2)}$$

Let  $t$  be an extended thread of that strategy, let  $\vec{e} \in E$  and let suppose that  $t_{|\iota^{(1)}}$  is winning on  $a[\vec{e}/\vec{x}] \neq b[\vec{e}/\vec{x}]$ . Since  $t$  is a thread,  $t_{|\iota^{(2)}}$  is a thread, and  $t_{|\iota^{(1)}} = t_{|\iota^{(2)}}$  is also a thread. Therefore we have  $t_{|\iota^{(1)}} \in \mathcal{W}_{a[\vec{e}/\vec{x}] \neq b[\vec{e}/\vec{x}]}$ . Since by hypothesis  $(a[\vec{e}/\vec{x}])^E = (b[\vec{e}/\vec{x}])^E$ , we have  $\mathcal{W}_{a[\vec{e}/\vec{x}] \neq b[\vec{e}/\vec{x}]} = \mathcal{W}_\perp$ , so  $t_{|\iota^{(2)}} = t_{|\iota^{(1)}} \in \mathcal{W}_\perp$ .  $\square$

The axioms for equality are:

$$\text{(refl)} \quad \forall x (x = x) \qquad \text{(Leib)} \quad \forall x \forall y (\neg A[x] \Rightarrow A[y] \Rightarrow x \neq y)$$

Recall that  $\forall x (x = x)$  is only syntactic sugar for  $\forall x (x \neq x \Rightarrow \perp)$ , and that  $\forall x \forall y (\neg A[x] \Rightarrow A[y] \Rightarrow x \neq y)$  is also syntactic sugar for:

$$\forall x \forall y ((A[x] \Rightarrow \perp) \Rightarrow A[y] \Rightarrow x \neq y)$$

**Lemma 15.** *Let  $\perp \subseteq E$ .*

1. *The strategy  $\llbracket \lambda x.x \rrbracket$  on  $\iota \rightarrow \iota$  is a realizer of (refl)*
2. *The strategy  $\llbracket \lambda x.x \rrbracket$  on  $(A^* \rightarrow \iota) \rightarrow A^* \rightarrow \iota$  is a realizer of (Leib)*

**Proof.**

1. This is a consequence of Lemma 14.

2. If  $t$  is an augmented thread of  $\llbracket \lambda x.x \rrbracket$  on the arena

$$\left( A^{*(1)} \rightarrow \iota^{(1)} \right) \rightarrow A^{*(2)} \rightarrow \iota^{(2)}$$

then  $t$  is even or infinite (since  $\llbracket \lambda x.x \rrbracket$  is total) and verifies  $t_{|_{A^{*(1)} \rightarrow \iota^{(1)}}} = t_{|_{A^{*(2)} \rightarrow \iota^{(2)}}}$ . Let  $e, f \in E$ . Suppose  $t_{|_{A^{*(1)} \rightarrow \iota^{(1)}}}$  is winning on  $\neg A[e]$  and  $t_{|_{A^{*(2)}}}$  is winning on  $A[f]$ . Since  $t$  is a thread,  $t_{|_{A^{*(2)} \rightarrow \iota^{(2)}}}$  is a thread so  $t_{|_{A^{*(1)} \rightarrow \iota^{(1)}}} = t_{|_{A^{*(2)} \rightarrow \iota^{(2)}}}$  is also a thread. This means that

$$t_{|_{A^{*(1)} \rightarrow \iota^{(1)}}} \in \mathcal{W}_{A[e] \rightarrow \perp}$$

We have to show then that  $t_{|\iota^{(2)}} \in \mathcal{W}_{e \neq f}$ . We distinguish two cases:

- $e \neq f$ : any position is in  $\mathcal{W}_{e \neq f}$  so in particular  $t_{|\iota^{(2)}} \in \mathcal{W}_{e \neq f}$
- $e = f$ : Since  $e = f$ ,  $\mathcal{W}_{A[e]} = \mathcal{W}_{A[f]}$ , and since  $t_{|_{A^{*(1)} \rightarrow \iota^{(1)}}} = t_{|_{A^{*(2)} \rightarrow \iota^{(2)}}}$ ,

$$\text{Threads} \left( t_{|_{A^{*(1)}}} \right) = \text{Threads} \left( t_{|_{A^{*(2)}}} \right)$$

so  $t_{|_{A^{*(1)}}}$  is winning on  $A[e]$ . Since  $t_{|_{A^{*(1)} \rightarrow \iota^{(1)}}} \in \mathcal{W}_{A[e] \rightarrow \perp}$ , we get  $t_{|\iota^{(1)}} \in \mathcal{W}_{\perp} = \mathcal{W}_{\perp}$ . Since  $e = f$ ,  $\mathcal{W}_{e \neq f} = \mathcal{W}_{\perp}$ , therefore

$$t_{|\iota^{(2)}} = t_{|\iota^{(1)}} \in \mathcal{W}_{e \neq f} \quad \square$$

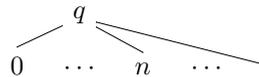
### 5. Peano arithmetic

We now proceed to the realizability interpretation of full Peano arithmetic.

#### 5.1. Definitions

Our first-order language is built from the function symbols  $0$  of arity 0,  $S$  of arity 1, and  $+$  and  $\times$  of arity 2. The only predicate symbol is  $\neq$  of arity 2. This choice of function symbols is only for simplicity, and we could choose to have all the symbols of primitive recursive functions.

We also fix the structure interpreting the terms of the logic to be the set of natural numbers  $\mathbb{N}$ . The symbols  $0$ ,  $S$ ,  $+$  and  $\times$  are interpreted the standard way. As stated in Sect. 4.1,  $\neq$  and  $\perp$  are both interpreted as  $\iota$ , and the associated arena in  $R^{\text{Fam}(C)}$  is  $\llbracket \iota \rrbracket = R^{(R^{\mathbb{N}})}$  where  $\mathbb{N} = \{U_n \mid n \in \mathbb{N}\}$ . Hence the type of natural numbers is interpreted as the negative translation of  $\mathbb{N}$ . Note that this is the usual flat arena of natural numbers:



This differs from Laird’s interpretation of PCF with control [16], where the base type of natural numbers is interpreted by the arena  $(\iota \rightarrow \iota) \rightarrow \iota$ .

#### 5.2. The relativization predicate

In the setting presented before, a realizer of  $\forall xA$  had to be uniform, meaning that it had to be a realizer of  $A[n/x]$  for any  $n \in \mathbb{N}$ . This choice suffices 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, we need to have a different notion of a realizer of  $\forall xA$ . Indeed, we realize the axiom scheme of induction using the recursor

of Gödel’s system T, which needs to know on which particular natural number to recurse. Therefore, we want a realizer of  $\forall xA$  to be a strategy that, given an element  $n \in \mathbb{N}$ , provides a realizer of  $A[n/x]$ . Instead of changing the realizability semantics of quantifiers, we add a syntactic construction for relativization. We follow [20] and extend the syntax of formulas by:

$$A, B ::= \dots \mid \{a\} \Rightarrow A$$

We use syntactic sugar:  $\forall^r xA \triangleq \forall x(\{x\} \Rightarrow A)$  and  $\exists^r xA \triangleq \neg \forall^r x \neg A$ . We extend our notion of context so  $\Gamma$  may now contain some first-order variables, along with formulas. These variables appearing in  $\Gamma$  are the relativized ones. To introduce and eliminate our new construct, we add the rules:

$$\frac{\Gamma, x \vdash A}{\Gamma \vdash \{x\} \Rightarrow A} \quad \frac{\Gamma \vdash \{a\} \Rightarrow A}{\Gamma \vdash A} \text{ (FV}(a) \subseteq \Gamma)$$

The translations of the new formulas are  $(\{a\} \Rightarrow A)^* = \iota \rightarrow A^*$ , and the associated winning conditions are  $\mathcal{W}_{\{a\} \Rightarrow A} = \mathcal{W}_{\{a^{\mathbb{N}}\} \rightarrow A}$ , which are defined since  $\{a^{\mathbb{N}}\} \subseteq \mathbb{N}$  (see Sect. 4.1). The ex-falso rule is still valid, thanks to the restriction of  $\{a\}$  being on the left of an arrow.

In order to define the translation of the new rules into  $\lambda\mu$ -calculus, we associate to each first-order term  $a$  with free variables  $x_1, \dots, x_n$  a  $\lambda\mu$  term  $a^*$  with free variables  $x_1^t, \dots, x_n^t$  and no free name. For that we assume a given constant  $k_f$  of type  $\iota \rightarrow \dots \rightarrow \iota \rightarrow \iota$  ( $n + 1$  times) for each function symbol  $f$  of arity  $n$ .  $a^*$  is then defined by induction:  $x^* = x^t$  and  $(f(a_1, \dots, a_n))^* = k_f a_1^* \dots a_n^*$ .

Since the context  $\Gamma$  may now contain first-order variables, we define a new mapping from first-order proofs to  $\lambda\mu$ -terms: if  $\nu$  is a derivation of:

$$x_1, \dots, x_m, A_1, \dots, A_n \vdash A$$

then  $\nu^*$  is a typed  $\lambda\mu$ -term of type  $A^*$  with free variables in  $x_1^t, \dots, x_m^t, x^{A_1^*}, \dots, x^{A_n^*}$ . The translation for the rules of the base system doesn’t change, and the new rules are translated to:

$$\frac{\frac{\nu}{\Gamma, x \vdash A}}{\Gamma \vdash \{x\} \Rightarrow A} \rightsquigarrow \lambda x. \nu^* \quad \frac{\frac{\nu}{\Gamma \vdash \{a\} \Rightarrow A}}{\Gamma \vdash A} \text{ (FV}(a) \subseteq \Gamma) \rightsquigarrow \nu^* a^*$$

We now define the interpretations of the constants  $k_f$  of  $\lambda\mu$ -calculus as strategies. In fact, we first define strategies for any function from  $\mathbb{N}^n$  to  $\mathbb{N}$  and then we use the interpretation  $f^{\mathbb{N}}$  to define  $\llbracket k_f \rrbracket$ .

In  $\text{Fam}(\mathcal{C})$  a morphism from  $\mathbf{N}^n = \{\mathcal{U}_{n_1} \times \dots \times \mathcal{U}_{n_n} \mid (n_1, \dots, n_n) \in \mathbb{N}^n\}$  to  $\mathbf{N} = \{\mathcal{U}_n \mid n \in \mathbb{N}\}$  is given by a function from  $\mathbb{N}^n$  to  $\mathbb{N}$  together with a strategy from  $\mathcal{U} \times \dots \times \mathcal{U} = \mathcal{U}$  to  $\mathcal{U}$ . Since there is only one such strategy, such a morphism is just given by a function from  $\mathbb{N}^n$  to  $\mathbb{N}$ . We will denote these morphisms  $\tau_g$  for  $g : \mathbb{N}^n \rightarrow \mathbb{N}$ . From  $\tau_g : \mathbf{N}^n \rightarrow \mathbf{N}$  we define  $\sigma_g : \iota \rightarrow \dots \rightarrow \iota \rightarrow \iota$  ( $n + 1$  times) by:

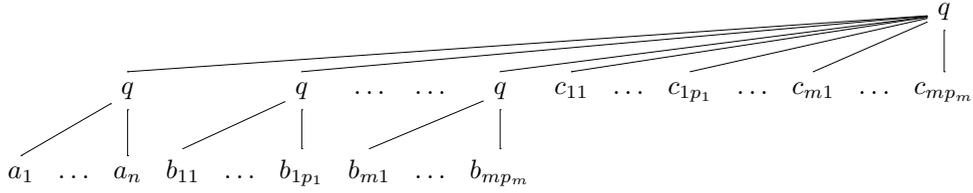
$$\sigma_g = \lambda x_1^t \dots \lambda x_n^t y^{R^{\mathbf{N}}} . x_1 (\lambda z_1^{\mathbf{N}} . x_2 (\lambda z_2^{\mathbf{N}} \dots \lambda z_n (\lambda z_n^{\mathbf{E}} . y (\tau_g z_1 \dots z_n))) \dots))$$

In particular, if  $n \in \mathbb{N}$  then  $\sigma_n$  is the strategy on  $\iota$  which answers  $n$  to the initial move. The following lemma states that the strategies  $\sigma_g$  indeed compute the correct values:

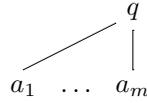
**Lemma 16.** *Let  $g : \mathbb{N}^n \rightarrow \mathbb{N}$  and  $n_1, \dots, n_n \in \mathbb{N}$ .  $\sigma_g$  is winning on:*

$$(\iota \rightarrow \dots \rightarrow \iota \rightarrow \iota, \mathcal{W}_{\{n_1\} \rightarrow \dots \rightarrow \{n_n\} \rightarrow \{g(n_1, \dots, n_n)\}})$$

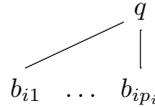
**Proof.** We prove the lemma for  $n = 2$  for clarity. In that case,  $\sigma_g$  is a strategy on  $\iota^{(1)} \rightarrow \iota^{(2)} \rightarrow \iota^{(3)}$ . Let  $t$  be an extended thread of  $\sigma_g$ , let  $n, n' \in E$  and let suppose that  $t_{|\iota^{(1)}}$  is winning on  $\mathcal{W}_{\{n\}}$  and  $t_{|\iota^{(2)}}$  is winning on  $\mathcal{W}_{\{n'\}}$ . The position corresponding to  $t$  has the following shape:



where  $c_{ij} = g(a_i, b_{ij})$ .  $t_{|\iota^{(1)}}$  is the following position:



Since  $t_{|\iota^{(1)}} \in \mathcal{W}_{\{n\}}$ , there is some  $i$  such that  $a_i$  is labeled with  $n$ . Since  $t_{|\iota^{(2)}}$  is winning on  $\mathcal{W}_{\{n'\}}$ , the following thread of  $t_{|\iota^{(2)}}$ :



is in particular in  $\mathcal{W}_{\{n'\}}$ , so there must be some  $j$  such that  $b_{ij}$  is labeled with  $n'$ . But then since  $c_{ij} = g(n, n')$ , we get that  $t_{|\iota^{(3)}} \in \mathcal{W}_{\{g(n, n')\}}$ .  $\square$

The constants  $k_f$  of  $\lambda\mu$ -calculus are then interpreted by  $\llbracket k_f \rrbracket = \sigma_{f^{\mathbb{N}}}$ . The formulation of the adequacy lemma in our new system reads as follows:

**Lemma 17.** *Let  $\perp \subseteq \mathbb{N}$ . Suppose that we have a realizer  $\llbracket k_A \rrbracket$  for each formula  $A \in \mathbf{Ax}$ . If  $\nu$  is a derivation of the sequent:*

$$x_1, \dots, x_m, A_1, \dots, A_n \vdash A$$

and if  $\theta$  is an assignment of elements of  $\mathbb{N}$  to free variables of  $A_1, \dots, A_n, A$ , then  $\llbracket \nu^* \rrbracket$  is a winning strategy on the arena:

$$\iota^{(1)} \rightarrow \dots \rightarrow \iota^{(m)} \rightarrow A_1^* \rightarrow \dots \rightarrow A_n^* \rightarrow A^*$$

equipped with winning condition:

$$\mathcal{W}_{\{\theta(x_1)\} \rightarrow \dots \rightarrow \{\theta(x_m)\} \rightarrow A_1[\theta] \Rightarrow \dots \Rightarrow A_n[\theta] \Rightarrow A[\theta]}$$

**Proof.** The case of the rules of the base system are unchanged. Therefore we prove the adequacy for the two new rules. For the introduction of relativization, the induction property is unchanged. For the elimination of relativization, let  $\nu$  be a proof of  $x_1, \dots, x_m, A_1, \dots, A_n \vdash \{a\} \Rightarrow A$  with  $\text{FV}(a) \subseteq \{x_1, \dots, x_m\}$ . By induction on  $a$ , using Lemma 16,  $\llbracket a^* \rrbracket$  is winning on:

$$\{\theta(x_1)\} \rightarrow \dots \rightarrow \{\theta(x_m)\} \rightarrow \left\{ (a[\theta])^{\mathbb{N}} \right\}$$

and we can conclude using Lemma 8.  $\square$

5.3. The realizability interpretation

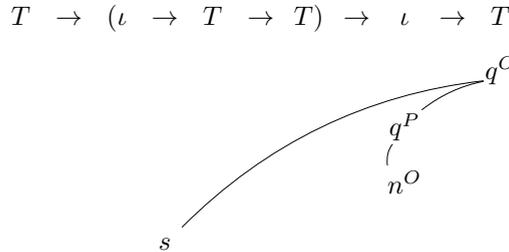
The winning conditions for  $\perp$  and  $e \neq f$  are as in Sects. 4.1 and 4.4. The axioms are the ones for equality (defined in Sect. 4.4) and:

$$\begin{array}{ll}
 (\mathbf{Snz}) & \forall x (S(x) \neq 0) \\
 (+\mathbf{0}) & \forall x (x + 0 = x) \\
 (+\mathbf{S}) & \forall x \forall y (x + S(y) = S(x + y)) \\
 (\mathbf{Sinj}) & \forall x \forall y (x \neq y \Rightarrow S(x) \neq S(y)) \\
 (\times \mathbf{0}) & \forall x (x \times 0 = 0) \\
 (\times \mathbf{S}) & \forall x \forall y (x \times S(y) = x + (x \times y)) \\
 (\mathbf{ind}) & \forall \vec{z} (A[0/x] \Rightarrow \forall^r x (A \Rightarrow A[S(x)/x]) \Rightarrow \forall^r x A)
 \end{array}$$

where  $A$  is a formula with free variables among  $x, \vec{z}$ . We will now define the realizers for these axioms. We first define  $\rho^T$ , the recursor on type  $T$ , which is the usual recursor of Gödel’s system T. For that we define for each  $n \in \mathbb{N}$  and simple type  $T$  a strategy  $\rho_n^T$  by:

$$\begin{aligned}
 \rho_0^T &= \llbracket \lambda xy.x \rrbracket : T \rightarrow (\iota \rightarrow T \rightarrow T) \rightarrow T \\
 \xi^T &= \llbracket \lambda nrxy.y n (r x y) \rrbracket : \iota \rightarrow (T \rightarrow (\iota \rightarrow T \rightarrow T) \rightarrow T) \rightarrow T \rightarrow (\iota \rightarrow T \rightarrow T) \rightarrow T \\
 \rho_{n+1}^T &= \xi^T (\sigma_n) (\rho_n^T) : T \rightarrow (\iota \rightarrow T \rightarrow T) \rightarrow T
 \end{aligned}$$

and we finally define the strategy  $\rho^T$  as the innocent strategy whose views are:



where  $q^O \overset{s}{\curvearrowright}$  is a view of  $\rho_n^T$  on the subarena  $T \rightarrow (\iota \rightarrow T \rightarrow T) \rightarrow T$ .

We use the following lemma in order to prove the validity of  $(\mathbf{ind})$ :

**Lemma 18.** *Let  $A$  be a formula with parameters and one free variable.*

1.  $\rho_0^{A^*}$  is a realizer of  $A[0] \Rightarrow \forall^r x (A[x] \Rightarrow A[S(x)]) \Rightarrow A[0]$
2.  $\xi^{A^*}$  is a realizer of:

$$\forall^r y ( (A[0] \Rightarrow \forall^r x (A[x] \Rightarrow A[S(x)]) \Rightarrow A[y]) \Rightarrow A[0] \Rightarrow \forall^r x (A[x] \Rightarrow A[S(x)]) \Rightarrow A[S y] )$$

**Proof.** This is an immediate consequence of Lemma 13, since the strategies  $\rho_0^{A^*}$  and  $\xi^{A^*}$  are the interpretations of proofs of the formulas.  $\square$

5.4. Validity of axioms

We prove that all the axioms are realized:

**Lemma 19.** *Let  $\perp \subseteq \mathbb{N}$ .*

1. *The empty strategy on  $\iota$  is a realizer of  $(\mathbf{S}nz)$*
2. *The strategy  $\llbracket \lambda x.x \rrbracket$  on  $\iota \rightarrow \iota$  is a realizer of  $(\mathbf{S}inj)$*
3. *The strategy  $\llbracket \lambda x.x \rrbracket$  on  $\iota \rightarrow \iota$  is a realizer of  $(+\mathbf{0})$ ,  $(\times\mathbf{0})$ ,  $(+\mathbf{S})$  and  $(\times\mathbf{S})$*
4.  *$\rho^{A^*}$  is a realizer of  $(\mathbf{ind})$*

**Proof.**

1. The only extended thread  $t$  of the empty strategy on  $\iota$  is the one-move thread. We have to show that for any  $n \in \mathbb{N}$ ,  $t \in \mathcal{W}_{\mathbf{S}(n) \neq 0}$ . Let  $n \in \mathbb{N}$ . Then we have  $(\mathbf{S}(n))^{\mathbb{N}} = n + 1 \neq 0 = \mathbf{0}^{\mathbb{N}}$  and  $\mathcal{W}_{\mathbf{S}(n) \neq 0}$  is the set of all positions on  $\iota$ . In particular  $t$ , which is the one-move position, is in  $\mathcal{W}_{\mathbf{S}(n) \neq 0}$ .
2. If  $t$  is an extended thread of  $\llbracket \lambda x.x \rrbracket$  on  $\iota^{(1)} \rightarrow \iota^{(2)}$ , then  $t$  is even or infinite (since the strategy is total), so  $t_{|\iota^{(1)}} = t_{|\iota^{(2)}}$ .  $t$  being a thread,  $t_{|\iota^{(2)}}$  is a thread, and  $t_{|\iota^{(1)}} = t_{|\iota^{(2)}}$  is also a thread. Let  $m, n \in \mathbb{N}$ . We have to show that if  $t_{|\iota^{(1)}} \in \mathcal{W}_{m \neq n}$ , then  $t_{|\iota^{(2)}} \in \mathcal{W}_{\mathbf{S}(n) \neq \mathbf{S}(m)}$ . We have  $m = n \Leftrightarrow (\mathbf{S}(n))^{\mathbb{N}} = (\mathbf{S}(m))^{\mathbb{N}}$  so  $\mathcal{W}_{m \neq n} = \mathcal{W}_{\mathbf{S}(n) \neq \mathbf{S}(m)}$  so we can conclude.
3. This is a consequence of [Lemma 14](#).
4. We first prove by induction on  $n \in \mathbb{N}$  that  $\rho_n^{A^*}$  is a realizer of the formula

$$A[0] \Rightarrow \forall^x x (A[x] \Rightarrow A[\mathbf{S}(x)]) \Rightarrow A[n]$$

where  $A$  is a formula with parameters and one free variable.

- The case for 0 is the first part of [Lemma 18](#)
- By induction hypothesis we have:

$$\rho_n^{A^*} \Vdash A[0] \Rightarrow \forall^x x (A[x] \Rightarrow A[\mathbf{S}(x)]) \Rightarrow A[n]$$

and by [Lemma 18](#) we have:

$$\xi^T \Vdash \forall^y y ((A[0] \Rightarrow \forall^x x (A[x] \Rightarrow A[\mathbf{S}(x)]) \Rightarrow A[y]) \Rightarrow A[0] \Rightarrow \forall^x x (A[x] \Rightarrow A[\mathbf{S}(x)]) \Rightarrow A[\mathbf{S}(y)])$$

so since  $\rho_{n+1}^T = \xi^T(\sigma_n)(\rho_n^T)$ , we get by [Lemmas 16 and 8](#):

$$\rho_{n+1}^{A^*} \Vdash A[0] \Rightarrow \forall^x x (A[x] \Rightarrow A[\mathbf{S}(x)]) \Rightarrow A[\mathbf{S}(n)]$$

which terminates the induction case since  $n + 1 = (\mathbf{S}(n))^{\mathbb{N}}$ .

Let now  $t$  be an augmented thread of  $\rho^{A^*}$  on the arena

$$A^{*(1)} \rightarrow (\iota^{(1)} \rightarrow A^{*(2)} \rightarrow A^{*(3)}) \rightarrow \iota^{(2)} \rightarrow A^{*(4)}$$

Let suppose that  $t_{|A^{*(1)}}$  is winning on  $A[0]$  and  $t_{|\iota^{(1)} \rightarrow A^{*(2)} \rightarrow A^{*(3)}}$  is winning on  $\forall^x x (A[x] \Rightarrow A[\mathbf{S}(x)])$ . We want to prove that  $t_{|\iota^{(2)} \rightarrow A^{*(4)}}$  is winning on  $\forall^x x A[x]$ , so let  $n \in \mathbb{N}$  and let suppose that  $t_{|\iota^{(2)}}$  is winning on  $(\iota, \mathcal{W}_{\{n\}})$ . Then there must be some  $n^O$  in  $t_{|\iota^{(2)}}$ . Let  $u$  be the subsequence of  $t$  consisting of the initial  $q^O$ , the following  $q^P$ , this  $n^O$  and all the moves of  $t$  such that the view obtained immediately after having been played contains  $n^O$ . Then  $u$  is a play of  $\rho_n^{A^*}$ . Since a  $P$ -move does not change the

current view, the positions of  $u_{|_{A^*(1)}}$  are  $O$ -subpositions of  $t_{|_{A^*(1)}}$  (the projection induces an inversion of polarities), so they are winning on  $A[0]$ , and the positions of  $u_{|_{L(1) \rightarrow A^*(2) \rightarrow A^*(3)}}$  are  $O$ -subpositions of  $t_{|_{L(1) \rightarrow A^*(2) \rightarrow A^*(3)}}$ , so they are winning on  $\forall^r x (A[x] \Rightarrow A[S(x)])$ . Then by the property on  $\rho_n^{A^*}$ ,  $u_{|_{A^*(4)}}$  is winning on  $A[n]$ . But  $u_{|_{A^*(4)}}$  is a  $P$ -subposition of  $t_{|_{A^*(4)}}$  (no inversion here), so  $t_{|_{A^*(4)}}$  is winning on  $A[n]$ .  $\square$

We define for a formula  $A$  of Peano Arithmetic its relativization  $A^r$  which is obtained by replacing every  $\forall$  by a  $\forall^r$ . The following result is easy to obtain, by replacing every rule about  $\forall$  by two rules: one about  $\forall$  and one about the relativization construct. It is also easy to derive  $A^r$  from  $A$  for each  $A \in \mathbf{Ax}$ , except the induction scheme, which is already relativized. Therefore we have the following result:

**Theorem 1.** *If  $A$  is provable in Peano arithmetic then  $A^r$  is provable in our system, so there is a computable strategy  $\sigma$  such that  $\sigma \Vdash A^r$ .*

### 5.5. Extraction

We now show that from any  $\Pi_2^0$ -formula provable in Peano arithmetic we can extract a computable witnessing function.

Suppose that we have a proof of  $\vdash \forall^r x \exists^r y (a = b)$ . We obtain by double-negation elimination a proof of  $\vdash \forall^r x (\neg \forall^r y (a \neq b))$ , and we map it to a strategy  $\sigma$  such that:

$$\sigma \Vdash \forall^r x (\neg \forall^r y (a \neq b)) \equiv \forall^r x (\forall^r y (a \neq b) \Rightarrow \perp)$$

Then if  $n \in \mathbb{N}$ , by [Lemmas 16 and 8](#),  $\sigma(\sigma_n) \Vdash \forall^r y (a[n/x] \neq b[n/x]) \Rightarrow \perp$ . Let now fix  $\perp = \{m \in \mathbb{N} \mid (a[n/x, m/y])^{\mathbb{N}} = (b[n/x, m/y])^{\mathbb{N}}\}$ . By a simple disjunction of cases we get

$$\llbracket \lambda x. x \rrbracket \Vdash \forall^r y (a[n/x] \neq b[n/x])$$

therefore by [Lemma 8](#),  $\sigma(\sigma_n) (\llbracket \lambda x. x \rrbracket) \Vdash \perp$ . Then  $\sigma(\sigma_n) (\llbracket \lambda x. x \rrbracket)$  is some  $\sigma_m$  such that  $m \in \perp$ . Indeed, if  $\sigma(\sigma_n) (\llbracket \lambda x. x \rrbracket)$  is the empty strategy then its only augmented play is  $q^O$ , which is losing on  $\perp$ .

## 6. Conclusion & future work

We would like to thank the anonymous reviewers for their helpful comments.

We have built a realizability model for Peano arithmetic using winning conditions on arenas, and have used it in the context of witness extraction for  $\Pi_2^0$ -formulas. Future work will be the comparison of the present model with the game interpretation of classical arithmetic of [\[7\]](#). Our main goal is to compare two different versions of realizers for the axiom of dependent choices: the modified bar recursion of [\[3\]](#) and the clock of [\[13\]](#).

### Acknowledgements

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] S. Abramsky, K. Honda, G. McCusker, A fully abstract game semantics for general references, in: 13th Annual IEEE Symposium on Logic in Computer Science, IEEE Computer Society, 1998, pp. 334–344.
- [2] S. Abramsky, G. McCusker, Call-by-value games, in: 6th EACSL Annual Conference on Computer Science Logic, in: *Lecture Notes in Computer Science*, Springer, 1997, pp. 1–17.
- [3] S. Berardi, M. Bezem, T. Coquand, On the computational content of the axiom of choice, *J. Symbolic Logic* 63 (2) (1998) 600–622.
- [4] V. Blot, Realizability for Peano arithmetic with winning conditions in HON games, in: 11th International Conference on Typed Lambda Calculi and Applications, in: *Lecture Notes in Computer Science*, vol. 7941, Springer, 2013, pp. 77–92.
- [5] P. Boudes, Thick subtrees, games and experiments, in: 9th International Conference on Typed Lambda Calculi and Applications, in: *Lecture Notes in Computer Science*, vol. 5608, Springer, 2009, pp. 65–79.
- [6] P. Clairambault, Least and greatest fixpoints in game semantics, in: 12th International Conference on Foundations of Software Science and Computational Structures, in: *Lecture Notes in Computer Science*, Springer, 2009, pp. 16–31.
- [7] T. Coquand, A semantics of evidence for classical arithmetic, *J. Symbolic Logic* 60 (1) (1995) 325–337.
- [8] T. Griffin, A formulae-as-types notion of control, in: 17th Symposium on Principles of Programming Languages, ACM Press, 1990, pp. 47–58.
- [9] R. Harmer, Games and full abstraction for non-deterministic languages, Ph.D. thesis, Imperial College London (University of London), 1999.
- [10] M. Hyland, Game semantics, in: A. Pitts, P. Dybjer (Eds.), *Semantics and Logics of Computation*. Publications of the Newton Institute, Cambridge University Press, 1997, pp. 131–184, Ch. 4.
- [11] M. Hyland, L. Ong, On full abstraction for PCF: I, II, and III, *Inform. and Comput.* 163 (2) (2000) 285–408.
- [12] J.-L. Krivine, Typed lambda-calculus in classical Zermelo–Fränkel set theory, *Arch. Math. Logic* 40 (3) (2001) 189–205.
- [13] J.-L. Krivine, Dependent choice, ‘quote’ and the clock, *Theoret. Comput. Sci.* 308 (1–3) (2003) 259–276.
- [14] J.-L. Krivine, Realizability in classical logic, *Panor. Synthèses* 27 (2009) 197–229.
- [15] J. Laird, Full abstraction for functional languages with control, in: 12th Annual IEEE Symposium on Logic in Computer Science, IEEE Computer Society, 1997, pp. 58–67.
- [16] J. Laird, A semantic analysis of control, Ph.D. thesis, University of Edinburgh, 1999.
- [17] P.-A. Melliès, Asynchronous games 3 an innocent model of linear logic, *Electron. Notes Theor. Comput. Sci.* 122 (2005) 171–192.
- [18] P.-A. Melliès, Sequential algorithms and strongly stable functions, *Theoret. Comput. Sci.* 343 (1–2) (2005) 237–281.
- [19] P.-A. Melliès, Asynchronous games 2: the true concurrency of innocence, *Theoret. Comput. Sci.* 358 (2–3) (2006) 200–228.
- [20] A. Miquel, Existential witness extraction in classical realizability and via a negative translation, *Log. Methods Comput. Sci.* 7 (2) (2011).
- [21] H. Nickau, Hereditarily sequential functionals, in: Third International Symposium on Logical Foundations of Computer Science, in: *Lecture Notes in Computer Science*, Springer, 1994, pp. 253–264.
- [22] P. Selinger, Control categories and duality: on the categorical semantics of the  $\lambda\mu$  calculus, *Math. Structures Comput. Sci.* 11 (2) (2001) 207–260.
- [23] A.S. Troelstra, Realizability, in: S. Buss (Ed.), *Handbook of Proof Theory*, in: *Studies in Logic and the Foundations of Mathematics*, vol. 137, Elsevier, 1998, pp. 407–473, Ch. 6.