Games and Definability for System F

Dominic J.D. Hughes
Oxford University Computing Laboratory
Wolfson Building
Parks Road
Oxford OX1 3QD
United Kingdom*
dhughes@comlab.ox.ac.uk

Abstract

We present a game-theoretic model of the polymorphic λ-calculus, system F, as a fibred category. Every morphism σ of the model defines an η-expanded, β-normal form $\hat{\sigma}$ of system F whose interpretation is σ. Thus the model gives a precise, non-syntactic account of the calculus.

1 Introduction

Polymorphism is of fundamental interest to computer scientists. Consider the function *map* in the setting of a typed functional programming language which takes a function $f : X \to Y$ and a list $a_1 a_2 \ldots a_n$ of elements each of type X and returns the list $f(a_1)f(a_2)\ldots f(a_n)$ of elements each of type Y. Since the algorithm is independent of the actual data types X and Y, it would be useful to have the flexibility of taking X and Y as type *variables*, to be instantiated whenever *map* is called. For example, we might write $map(Nat)(Bool)$ to mean "the *map* function as defined above, with $X := Nat$ and $Y := Bool$", thus extending the notion of application from the usual form *term(term)* to the form *term(type)*. The function *map* is said to be *polymorphic*, and is assigned the type

$$T = \Pi X.\, \Pi Y.\, ((X \to Y) \to (\text{list}X \to \text{list}Y)).$$

The notation is designed according to the intuition that this type is a "product" indexed by X and Y ranging over all possible types.

A key tool for the analysis of polymorphism is system F [Gir89], which is the simply-typed λ-calculus extended by quantification over type variables, as above. Models of system F do not come about easily. The impredicative nature of the definition of T, that T itself is in the range of X and Y, raises problems for the semanticist. The two standard classes of models are the *PER* models [Cro93], and the domain models [CGW89].

Recent advances in our understanding of the nature of sequential computation were made through the paradigm of modelling *term(term)* application *intensionally*, by interaction between function and argument. Interaction occurs by the repeated exchange of basic data "tokens" (for example natural numbers, booleans, and requests for data). Thus higher-order functions do not necessarily have instant access to the full information of the whole graph of an argument function f: information can be obtained only by testing the input-output behaviour of f, as a "black box". This approach forms the basis of *game semantics*, and has led to many good results, for example the solution of the long-standing full abstraction problem for *PCF* [AJM94, HO94, Nic94].

In this paper we present a games model for

*This is a slightly revised version of a paper appearing in *IEEE 12th Symposium on Logic in Computer Science*, Warsaw, June 1997.

system F. A games model for polymorphism was presented in [Abr95]. The new concept in our approach is that we model not only $term(term)$ application intensionally, but also $term(type)$ application; interaction occurs not only through the communication of basic data tokens, but also by the direct exchange of types themselves. A type is modelled as a ***first-order polymorphic arena***, the analogue for system F of the computational arena of Hyland and Ong for *PCF* [HO94][1]. A ***second-order polymorphic arena*** is an arena which has first-order polymorphic arenas as *moves*. The HO notion of a justified sequence of moves is enriched to a structure which we call a ***located*** sequence. Instead of the whole justified sequence being inside one particular arena, each move is located in a *different* arena. We use a special control move $\star$, called the ***initialising*** move, to open a new first-order thread of play *inside* one of the previous second-order moves of the sequence.

Our main result is that every strategy σ of the games model $\mathbb{F}$ defines an η-expanded, β-normal form $\hat{\sigma}$, whose interpretation is σ. This provides us with a very precise non-syntactic characterisation of system F.

2 Syntax of system F

We fix the notation of system F. Types are generated by

$$T \ ::= \ X \mid unit \mid T \times T \mid T \to T \mid \Pi X.T$$

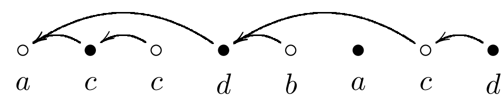
where X ranges over a countably infinite set of type variables. Raw terms are generated by

$$\begin{array}{rcl} s & ::= & x \mid \langle\rangle \mid \langle s, s\rangle \mid Fst(s) \mid Snd(s) \\ & & \mid \lambda x^T.s \mid ss \mid \Lambda X.s \mid sT \end{array}$$

where x is any term variable, and T is any type. See [Cro93] for further details of the language.

3 Games

We consider alternating, two-player games between P (think of Player, Program, or System) and O (Opponent, User, or Environment) in which P starts. The following diagram represents a trace or history of events in a game:



The labels a, b, c and d are actions names, just like the actions of a labelled transition system. Each node signifies an occurrence of an action, and time runs from left to right. A hollow node $\circ$ indicates that an action was performed by P, and a solid node $\bullet$ indicates that an action was performed by O. Arcs pointing backwards in time indicate a causal relationship between actions, for example P's action c occurring in seventh position was justified by O's action d occurring in fourth position.

We formalise these concepts in the definitions below. A ***justified sequence of events*** s on the set of actions Act is a structure $\langle E_s, <_s, \mathsf{just}_s, \mathsf{pl}_s, \mathsf{act}_s \rangle$ consisting of:

- *Events.* A finite set $E_s = \{x, y, z, \ldots\}$ of ***events***, pictured as nodes in the diagram above.

- *Temporal relation.* A strict linear ordering $<_s$ on events. Denote the corresponding successor partial function by $\mathsf{succ}_s : E_s \rightharpoonup E_s$.

- *Justification pointer.* A partial function $\mathsf{just}_s : E_s \rightharpoonup E_s$. Define the relation y is ***justified*** by x, written $x \curvearrowleft_s y$, if and only if $\mathsf{just}_s(y) = x$.

- *Association with players.* A total function $\mathsf{pl}_s : E_s \to \{\mathsf{O}, \mathsf{P}\}$ associating each event with a particular player.

- *Labelling with actions.* A total function $\mathsf{act}_s : E_s \to \mathsf{Act}$, associating an action to each event.

The structure is required to satisfy the following conditions:

J1. *Causality.* Justification points backwards in time: $x \curvearrowleft_s y$ only if $x <_s y$.

[1] In future we abbreviate this reference to "HO".

J2. P *goes first.* If x is the first event in s then $\mathsf{pl}_s(x) = \mathsf{P}$.

J3. *Alternating play.* The players take turns, and justification has to point to an event associated with the opposite player:

   (a) If $\mathsf{succ}_s(x) = y$, then $\mathsf{pl}_s(x) \neq \mathsf{pl}_s(y)$.

   (b) If $x \curvearrowleft_s y$, then $\mathsf{pl}_s(x) \neq \mathsf{pl}_s(y)$.

Two justified sequences of events s and t are ***isomorphic*** if there exists a bijection $\theta : E_s \to E_t$ of the underlying sets of events that preserves and reflects structure. We say s is a ***subsequence*** of t if there exists an injection $i : E_s \to E_t$ that preserves and reflects structure, and s is a ***prefix*** of t if in addition the image of E_s is an initial segment of the events of t. The empty justified sequence of events ε is defined by $E_\varepsilon = \emptyset$. A ***justified sequence*** is an isomorphism class of justified sequences of events.

A ***move*** is the information content of the "difference" between two consecutive justified sequences, *i.e.* a pair (a, i) where a is an action and i is either an odd natural number indicating how far back to justify, or $i = 0$ meaning no pointer. In this way any justified sequence can be encoded as a string of moves, $(a, i)(b, j)(c, k) \ldots \in (\mathsf{Act} \times \mathbb{N})^*$. This alternative representation is convenient for manipulations of justified sequences that are "local", such as in the definition of strategy below, where we are concerned only with the very end of a justified sequence. The original notation is useful for "global" operations such as the deletion of events from a justified sequence.

Notation We take $a, b, \ldots$ to range over actions, $x, y, \ldots$ to range over events, $s, t, \ldots$ to range over justified sequences, and $m, n, \ldots$ to range over moves. Furthermore we write x^a to indicate that the event x is labelled with the action a. We will identify a justified sequence s with its encoding as a string of moves. Concatenation of moves and of strings of moves is represented by juxtaposition. If it is clear from the context that an action a occurs at a particular event x, for example in a string of moves smn where $m = (a, i)$, we will often abuse notation and simply write the action a or the move m to mean the event x.

3.1 Games, strategies, and arenas

A game is a specification of certain allowable strings of moves. For example, the game of chess, with moves such as "Knight $f3-g5$", can be specified abstractly as the set of all strings of moves that follow the rules. Formally, a ***game*** G is a pair $\langle \mathsf{Act}_G, L_G \rangle$ consisting of a set of actions Act_G, together with a non-empty, prefix-closed set L_G of justified sequences on Act_G, called the ***legal sequences*** of G.

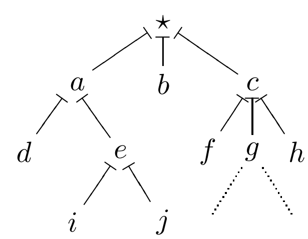
A strategy on a game is a rule telling P what to do given any reachable[2] legal sequence in which it is his turn to play next. It is best thought of as a function that maps a reachable sequence s to a move m_s, such that sm_s is (the move-encoding of) a legal sequence.

A ***partial strategy*** σ on a game G is a non-empty, prefix-closed set of legal sequences $\sigma \subseteq L_G$ satisfying

- P-*determinism.* Whenever $sm, sn \in \sigma$ with $\mathsf{pl}_{sm}(m) = \mathsf{pl}_{sn}(n) = \mathsf{P}$, then $m = n$.
- O-*contingent completeness.* If $sm \in L_G$ with $\mathsf{pl}_{sm}(m) = \mathsf{O}$ and $s \in \sigma$, then $sm \in \sigma$.

A ***winning sequence*** is a legal sequence which is prefix-maximal in L_G and in which P performed the last action, and a ***winning strategy*** is a partial strategy that is the prefix-closure of a set of winning sequences.

One way to specify the set of legal sequences of a game is to enforce a particular causal relationship between the actions. An ***arena*** is a pair $\langle \mathsf{Act}_A, \vdash_A \rangle$ where Act_A is a set of actions and $\vdash_A$, the ***enabling*** relation of A, is a binary relation between $\mathsf{Act} + \{\star\}$ ("+" denotes disjoint union) and Act such that for each $a \in \mathsf{Act}$ there exists a unique finite path $\star \vdash \ldots \vdash a$.



[2] The domain of definition need only be sequences reached according to the previous play dictated by the strategy.

The direct descendents of the ***initialising action*** $\star$ are called ***initial actions***, which in the graph above are the actions a, b and c. An A***-sequence*** is a justified sequence s on the set of actions $\mathsf{Act} + \{\star\}$ such that

1. Justification respects enabling: $x^a \curvearrowleft_s y^b$ only if $a \vdash_A b$.
2. Every non-initialising event is justified: for any event y^b with $b \neq \star$ there exists an event x such that $x \curvearrowleft_s y^b$.

Any non-empty A-sequence necessarily begins with an unjustified initialising action by P. An A***-game*** is a game $\langle\, \mathsf{Act}_A + \{\star\},\, L_G \,\rangle$ for which L_G consists only of A-sequences.

The next section highlights how the structures defined so far relate to previous work on game semantics, and can skipped by a reader steering a direct course for system F.

3.2 Relationship with HO-games

A justified sequence s is a P***-view*** if justification by O is only ever to the immediately preceding P-event. Define the P***-view game*** on an arena A to be the maximal A-game whose set of legal sequences consists only of P-views, and in which there is at most one occurrence of the initialising action $\star$. Define a ***starting strategy*** to be a partial strategy in which P successfully performs his first $\star$ action, *i.e.* which is not equal to the singleton set consisting of the empty sequence.

Proposition 1 *(HO-correspondence.) Starting strategies on* P*-view games correspond to innocent strategies on answer-free HO-arenas.*

The proof is trivial: actions correspond to questions, and given a legal sequence on a P-view game, delete the P-action $\star$ from the front in order to obtain an HO-P-view.

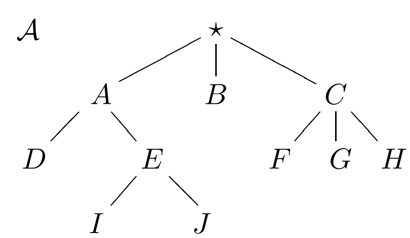
4 Second-order structure

A ***universe*** of actions is a set $\mathcal{U}$ such that $\mathcal{U} + \mathcal{U} \hookrightarrow \mathcal{U}$ and $\mathcal{U} \times \mathcal{U} \hookrightarrow \mathcal{U}$. Let R^* denote the reflexive, transitive closure of a binary relation R, and let $\mathbb{N}^+ = \mathbb{N} \setminus \{0\}$. A ***polymorphic arena*** A on the universe $\mathcal{U}$ is a quadruple $\langle\, \mathsf{Act}_A,\ \vdash_A,\ \mathsf{ar}_A,\ \mathsf{ref}_A \,\rangle$ consisting of

- An arena $\langle\, \mathsf{Act}_A,\ \vdash_A \,\rangle$ such that $\mathsf{Act}_A \subseteq \mathcal{U}$.
- A function $\mathsf{ar}_A : \mathsf{Act}_A \to \mathbb{N}$ assigning an ***arity*** to each action a. By convention $\mathsf{ar}_A(\star) = \infty$.
- A function $\mathsf{ref}_A : \mathsf{Act}_A \to (\mathsf{Act}_A + \{\star\}) \times \mathbb{N}^+$ assigning a ***reference*** further up the $\vdash_A$-tree: if $\mathsf{ref}_A(a) = (b, i)$ then $i \leq \mathsf{ar}_A(b)$ and $b \vdash_A^* a$. We write b_i for (b, i).

A selection of polymorphic arenas are shown in Figure 1. If non-zero, the arity of an action is placed to its left; the reference is placed in brackets to the right. We take $A, B, C, \ldots$ to range over polymorphic arenas, and the set of polymorphic arenas on $\mathcal{U}$ is denoted $\mathbb{PA}(\mathcal{U})$. Take the judgement $n \vdash A$ to mean that $i \leq n$ for each reference $\star_i$ of A, in which case we say A is ***in context*** n. (The overloading of the symbol $\vdash$ should not cause any confusion.)

A ***second-order polymorphic arena*** $\mathcal{A}$ over the universe $\mathcal{U}$ is a polymorphic arena on the universe $\mathbb{PA}(\mathcal{U})$. We take $\mathcal{A}, \mathcal{B}, \mathcal{C} \ldots$ to range over second-order polymorphic arenas, and the set $\mathbb{PA}(\mathbb{PA}(\mathcal{U}))$ of second-order polymorphic arenas over $\mathcal{U}$ will be abbreviated to $\mathbb{PA}^{\nvDash}(\mathcal{U})$.



A polymorphic arena $A \in \mathbb{PA}(\mathcal{U})$ will be called a ***1-arena***, and its actions $a \in \mathcal{U}$ called ***1-actions***. A second-order polymorphic arena $\mathcal{A} \in \mathbb{PA}^{\nvDash}(\mathcal{U})$ will be called a ***2-arena***, and its actions $A \in \mathbb{PA}(\mathcal{U})$ called ***2-actions***. Thus 2-actions are 1-arenas. By convention $\star$ is both a 1-action and a 2-action.

4.1 Located sequences

A (second-order) ***located sequence*** s in a universe $\mathcal{U}$ of actions is a structure $\langle E_s, <_s, \mathsf{just}_s, \mathsf{pl}_s, \mathsf{act}_s, \mathsf{loc}_s \rangle$ consisting of

- A justified sequence on the set of actions $\mathcal{U} + \mathbb{PA}(\mathcal{U}) + \{\star\}$. In other words act_s assigns either a 1-action or a 2-action to each event.
- A function $\mathsf{loc}_s : E_s \to \mathbb{PA}(\mathcal{U}) + \mathbb{PA}^{\nvDash}(\mathcal{U})$ setting the ***location*** of each event.

The structure is required to satisfy the following conditions:

L1. Events are well-located: for all events x^α of s, we have $\alpha \in \mathsf{Act}_{\mathsf{loc}_s(x^\alpha)} + \{\star\}$.

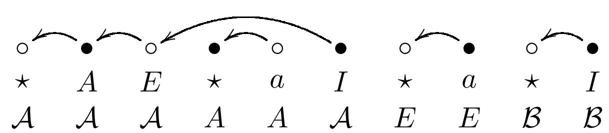
L2. Justification respects location: if $x \curvearrowleft_s y$ then $\mathsf{loc}_s(x) = \mathsf{loc}_s(y)$.

L3. Justification respects enabling locally: if $x \curvearrowleft_s y$ then $x \vdash_{\mathsf{loc}_s(x)} y$.

L4. Demands for the other player to play an initial move on a new arena are satisfied uniquely and immediately: if $\mathsf{act}_s(x) = \star$ then $x \curvearrowleft_s y$ if and only if $\mathsf{succ}_s(x) = y$.

Define $x \in E_s$ to be a ***1-event*** or a ***2-event*** according as the location of x is a 1-arena or a 2-arena.

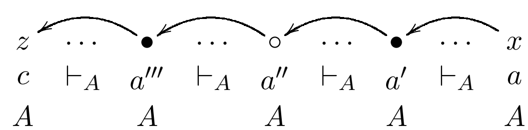
A typical located sequence is depicted below.



The first action is by P, initialising the 2-arena $\mathcal{A}$ which was pictured above. This forces O to play an initial action in $\mathcal{A}$, and she chooses the 2-action A, a 1-arena. P continues inside $\mathcal{A}$ by responding to A with the 1-arena E. Now O initialises the 1-arena A, which occurred earlier as a 2-action in the sequence (the restriction to a previous 2-action is not a consequence of the conditions above, but will be required later). P obliges with an initial 1-action a inside A. Justifying back to the original thread of play inside the 2-arena $\mathcal{A}$, O continues with the 2-action I. Now P forces O to play an initial 1-action in E by locating the initialising move $\star$ in E, a previous 2-action in the sequence. And so on. Note that the 1-action a occurs both as a P-action and as an O-action, and located inside two different arenas.

We end this section with a relationship between the enabling structure of a 1-arena and the justification structure of a located sequence.

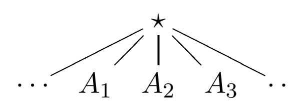
Lemma 1 *Let s be a located sequence. Suppose x^a is a 1-event of s located in the 1-arena A, and that $c \vdash_A^* a$. Then there exists a unique 1-event z^c of s such that $z^c \curvearrowleft_s^* x^a$.*



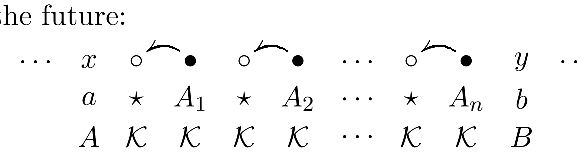
Proof: The existence and uniqueness of z^c comes from conditions L1-3 and the fact that the enabling relation forms a tree. □

5 Games for system F

We shall require only one particular 2-arena, a "flat" 2-arena with actions the set of all polymorphic arenas. We denote this 2-arena by $\mathcal{K}$, for *"Kind"*, as its elements are *"types"*:



Whenever a player performs a 1-event x^a with arity n (*i.e.* the arity of a is n inside the location 1-arena of x) it will be followed by $2n$ actions located in $\mathcal{K}$. Because of condition L4 and the fact that $\mathcal{K}$ is flat, this means we are forced to "import" n new 1-arenas which may become initialised in the future:



We will be considering only P-views as the projection used later for interaction is quite complex. Furthermore it is easier to extract definability from a standard notion of strategy on a P-view rather than the corresponding notion of a "second-order" innocent strategy on the total view (compare with Proposition 1). The component additional to HO-projection is that the collapse to a P-view involves the uniformity or type-independence associated with polymorphism: P has to proceed without the knowledge of which particular 1-arenas A were imported by O as 2-events located in $\mathcal{K}$. In going to the P-view we package all 2-actions $A, B, C, \ldots$ by O into a special symbol $\boldsymbol{?}$, indicating a parcel whose contents shall remain forever unknown to P. For example in the P-view the actions of the sequence displayed above would become $a \star \boldsymbol{?} \star \boldsymbol{?} \ldots \star \boldsymbol{?}\, b$. Secondly the fact that in the full view of the history of interaction these 2-actions may become initialised is also not visible in the P-view. Intuitively this is uniformity again: if we cannot see the 1-arena A hidden inside the parcel $\boldsymbol{?}$, we certainly cannot observe play inside A.

In system F there is a dependency of first-order entities (terms) on second-order entities (types). In a located sequence the dependency is achieved via a pointer from 1-events to previous 2-events, regulating the way in which new locations become initialised during the game.

Finally, P is constrained by a "copy-cat" condition, related to the notion of a copy-cat strategy, which is a standard concept in game semantics. The intuition is that during the composition (interaction) of strategies, moves will be played by P (unknowingly) inside the parcels of O. Since P is unaware of the contents of the $\boldsymbol{?}$-parcels, the only safe way of achieving this is by playing uniformly, copying moves between different threads of play in the same parcel. As soon as a parcel is "instantiated" to an arena B (corresponding to the substitution of a type $\hat{B}$ for a type variable — see section 6), the copy-cat condition becomes "instantiated" to a copy-cat strategy between occurences of B.

5.1 Conditions on located sequences

We specify eight conditions on a located sequence s that will be required in order to model system F. First adjoin the symbol $\boldsymbol{?}$ to the set of 1-arenas $\mathbb{PA}(\mathcal{U})$. Let $\mathcal{K} = \langle\, \mathsf{Act}_A,\ \vdash_A,\ \mathsf{ar}_A,\ \mathsf{ref}_A \,\rangle$ be the 2-arena given by $\mathsf{Act}_\mathcal{K} = \mathbb{PA}(\mathcal{U})$, $A \vdash_\mathcal{K} B$ if and only if $A = \star$ and $B \in \mathbb{PA}(\mathcal{U})$, $\mathsf{ar}_\mathcal{K}(A) = 1$, and $\mathsf{ref}_\mathcal{K}(A) = (A, 1)$. (Arity and reference are in fact arbitrary as this structure on $\mathcal{K}$ is never used.)

$F1$. ***The only 2-arena is*** $\boldsymbol{\mathcal{K}}$***.*** Whenever $\mathsf{loc}_s(x) \in \mathbb{PA}^{\nvDash}(\mathcal{U})$, $\mathsf{loc}_s(x) = \mathcal{K}$.

$F2$. ***Importation of new 1-arenas.*** Every non-initialising 1-event x^a is followed immediately by $2n$ 2-events, where n is the arity of a ($\mathsf{ar}_{\mathsf{loc}_s(x^a)}(a)$), and then by a 1-event y^b. (Unless s terminates beforehand.) See the diagram above.

$F3$. ***First-order*** P***-view.*** O's pointers do not skip over any of her own actions apart from initialisations: if $x \curvearrowleft_s y$ and $\mathsf{pl}_s(y) = \mathsf{O}$, then whenever $x <_s z^\alpha <_s y$ with $\mathsf{pl}_s(z^\alpha) = \mathsf{O}$, $\alpha = \star$.

$F4$. ***Second-order*** P***-view.*** For any 2-event x^A, $A = \boldsymbol{?}$ if and only if $\mathsf{pl}_s(x^A) = \mathsf{O}$. For any 1-event x^a, $\mathsf{loc}_s(x^a) \neq \boldsymbol{?}$.

$F5$. ***Contexts.*** Let n be the number of occurrences of $\boldsymbol{?}$ preceding the 2-event x^A, where $A \neq \star$ and $A \neq \boldsymbol{?}$. Then $n \vdash A$. Let m be the number of occurrences of $\boldsymbol{?}$ preceding the first 1-event $x^\star$ of s, and let A be the location of $x^\star$. Then $m \vdash_s A$.

$F6$. ***Locations.*** Let A be the location of the first 1-event $x^\star$ of s. Then every 1-event of s is located either in A or in the 2-action associated with an earlier 2-event.

The following is the formalisation of the dependency of 1-events on 2-events as motivated in the informal discussion. Let x^a be a 1-event of s located inside a 1-arena A, and suppose it is not the last 1-event of s. Let $c_i = \mathsf{ref}_A(a)$, the reference of the action a inside A.

1. If $c \neq \star$ then let z^c be as defined by Lemma 1, and let $\hat{z}^{A_i}$ be the $(2i)^{th}$ 2-event after z^c in s. (By condition $F2$ and the fact that x^a is not the last 1-event, s is of sufficient length for $\hat{z}^{A_i}$ to exist.)

2. If $c = \star$ then let the 2-event $w^{?}$ be the i^{th} occurrence of O's 2-action *?* in s. (By conditions $F5$ and $F6$, s has at least i occurrences of *?*.)

Define the ***2-reference*** of x^a in s by

$$\mathsf{ref}^2_s(x^a) \quad = \quad \begin{cases} \hat{z}^{A_i} & \text{if } c \neq \star, \\ w^{?} & \text{if } c = \star. \end{cases}$$

To understand case 1, insert the following segment in place of z in the diagram following Lemma 1:

$$\begin{array}{ccccccccc} z & \bullet & \circ & \cdots & \hat{z} & \cdots & \bullet & \circ \\ c & \star & A_1 & \cdots & A_i & \cdots & \star & A_n \\ A & \mathcal{K} & \mathcal{K} & & \mathcal{K} & & \mathcal{K} & \mathcal{K} \end{array}$$

Note that $A_i = ?$ if the O and P nodes are the other way round.

*F*7. ***Initialisation of locations.*** Let x^a be the most recent 1-event before the 1-event y^b, and let $\tilde{x}^B = \mathsf{ref}^2_s(x^a)$, the 2-reference of x^a. If $B \neq ?$ then $b = \star$ and $\mathsf{loc}_s(y^b) = B$. If $B = ?$ then $b \neq \star$ (consequently y^b is a justified event in one of the old locations).

$$\begin{array}{cccccccccccc} \cdots & \tilde{x} & \cdots & x & \circ & \bullet & \cdots & \circ & \bullet & y & \cdots \\ & B & & a & \star & A_1 & \cdots & \star & A_n & \star & \\ & \mathcal{K} & & A & \mathcal{K} & \mathcal{K} & \cdots & \mathcal{K} & \mathcal{K} & B & \end{array}$$

*F*8. ***Copy-cat.*** Whenever P makes a "hidden" 2-reference it must be the same as the most recent "hidden" 2-reference made by O. Suppose x^a is a non-initialising 1-event associated with P, and $\mathsf{act}_s(\mathsf{ref}^2_s(x^a)) = ?$. Let z^c be the most recent 1-event before x^a such that $\mathsf{act}_s(\mathsf{ref}^2_s(z^c)) = ?$. Then $\mathsf{ref}^2_s(x^a) = \mathsf{ref}^2_s(z^a)$.

5.2 The game associated with a 1-arena in context

Given a 1-arena in context $n \vdash A$, let $start(A, n)$ be the located sequence consisting of $2n$ actions located in $\mathcal{K}$ followed by the initialisation of A:

$$\begin{array}{cccccccc} \circ & \bullet & \circ & \bullet & \cdots & \circ & \bullet & \circ \\ \star & ? & \star & ? & \cdots & \star & ? & \star \\ \mathcal{K} & \mathcal{K} & \mathcal{K} & \mathcal{K} & \cdots & \mathcal{K} & \mathcal{K} & A \end{array}$$

Let $L_{A,n}$ be the set of located sequences in $\mathcal{U}$ satisfying conditions F1-8 above that are prefixes of or are prefixed by $start(A, n)$. Define $\underline{L}_{A,n}$ to be the set of justified sequences on $\mathcal{U} + \mathbb{PA}(\mathcal{U}) + \{\star\}$ obtained by forgetting location structure (in general a non-injective operation).

Lemma 2 $\underline{L}_{A,n} \cong L_{A,n}$

Proof: Whenever a justified move is made the location is determined by L2, and whenever an initialising move is made, the location is determined by $F7$. □

Thus by identifying $L_{A,n}$ with $\underline{L}_{A,n}$ we are able to define the ***game*** $G(A, n)$ ***associated with*** $n \vdash A$ to be $\langle \mathcal{U} + \mathbb{PA}(\mathcal{U}) + \{\star\}, L_{A,n} \rangle$, within the scope of the definition of section 3.1.

5.3 Examples

We examine some play on the 1-arenas depicted in Figure 1 and consider the winning strategies available on each. Below are the system F encodings of the interpreted types:

$$\begin{aligned} Emp &= \Pi X. X, \\ Sgl &= \Pi X. X \to X, \\ Bool &= \Pi X. X \to X \to X, \\ Nat &= \Pi X. X \to (X \to X) \to X. \end{aligned}$$

The game $G([\![\, unit \,]\!], 0)$**.** The only possible action is the initialisation of the game by P. Thus there is a unique winning strategy.

The game $G([\![\, Emp \,]\!], 0)$**.** This time O has the action a to continue with after the initialisation. Since the arity of a is 1, by condition $F2$ there must follow two actions inside $\mathcal{K}$. Since P cannot respond after this there are no winning strategies. ($F7$ forces him to play a justified move: he cannot justify from a because there is no action below a in the tree).

The game $G([\![\, Sgl \,]\!], 0)$**.** The first four moves are as for $[\![\, Emp \,]\!]$. This time P can justify back to the original thread of play inside $[\![\, Sgl \,]\!]$, playing c. The game is over, as there are no more actions beneath c in the tree and $F7$ forces O to play a justified move. Thus there is exactly one total strategy for P. The copy-cat condition $F8$ is satisfied as the 2-reference of P's action c is the

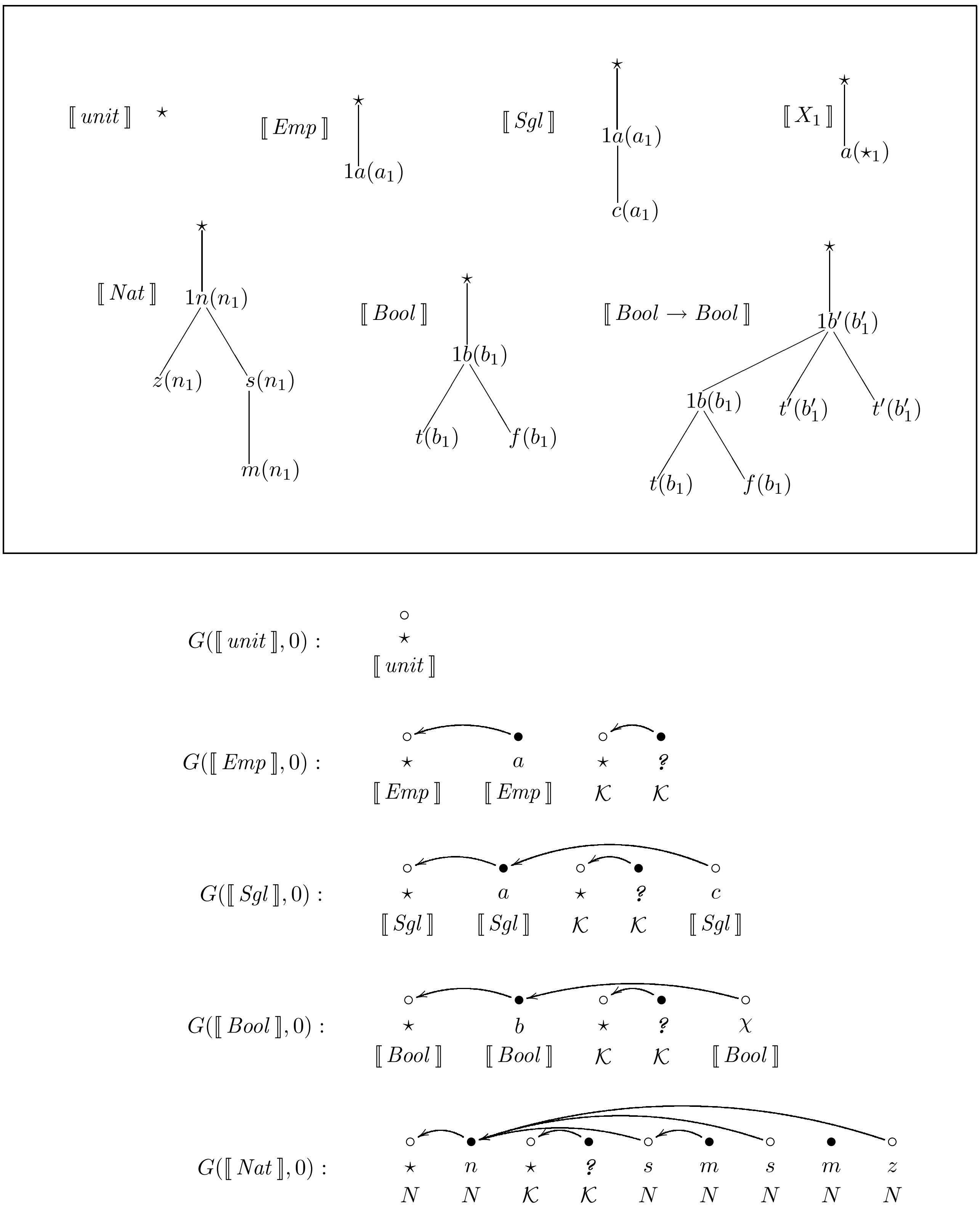


Figure 1. A selection of 1-arenas and located sequences.

fourth event, the same 2-reference as O's (most recent and only) 1-action a.

The game $G(\llbracket Bool \rrbracket, 0)$. This game is just like $G(\llbracket Sgl \rrbracket, 0)$ but for the fact that P now has the choice of the winning action as either $\chi = t$ or $\chi = f$, so there are two total strategies. Think of b as a request for boolean data by O, and t or f as the reply of true or false. These two winning strategies are the interpretation of the terms

$$\begin{aligned} tt &= \Lambda X.\lambda t^X.\lambda f^X.t, \\ ff &= \Lambda X.\lambda t^X.\lambda f^X.f, \end{aligned}$$

which are the system F encodings of true and false.

The game $G(\llbracket Nat \rrbracket, 0)$. Displayed is the winning position of the strategy interpreting the term $\Lambda X.\,\lambda z^X.\lambda s^{X\to X}.ssz$, which is the encoding of the natural number 2. We abbreviate $N = \llbracket Nat \rrbracket$. Think of the action n as a request for a number by O, z as zero by P, s as demand for output from the successor function by P, and m as a request for input by O to the successor function.

The game $G(\llbracket Bool \to Bool \rrbracket, 0)$. A type quantifier occurs on the left of an arrow in the type $Bool \to Bool$, and so we finally witness the full second-order machinery of "types as moves". We consider the interpretation of one particular system F encoding of the *not* function,

$$not = \lambda b^{Bool}.bBool\,ff\,tt.$$

Figure 2 shows one of its four winning positions on $G(\llbracket Bool \to Bool \rrbracket, 0)$. We abbreviate $B = \llbracket Bool \rrbracket$, $B^B = \llbracket Bool \to Bool \rrbracket$, and $X = \llbracket X_1 \rrbracket$. The corresponding sequence of regulatory conditions is: $start(B^B, 0)$, L4, $F2$, $F2$ and L4, P choice of b, $F2$, $F2$ and L4 and P choice of importing the arena B, $F7$, L4, $F2$, $F2$ and L4 and P choice of importing X, $F7$, L4, O choice of t inside B^B, $F7$, L4, $F2$, $F2$ and L4, P choice of f inside B.

6 Structure on $\mathbb{PA}(\mathcal{U})$

We define the product $A \times B$, function space $A \Rightarrow B$, universal quantification $\forall_n(A)$, and substitution of 1-arenas, and an equivalence of 1-arenas "up to renaming of actions",

Product. $A \times B$ is obtained by laying the $\vdash$-trees side by side and then identifying the two copies of $\star$:

$$\begin{aligned} \mathsf{Act}_{A\times B} &= \mathsf{Act}_A + \mathsf{Act}_B, \\ a \vdash_{A\times B} b &\iff a \vdash_A b \text{ or } a \vdash_B b, \\ \mathsf{ar}_{A\times B} &= [\mathsf{ar}_A, \mathsf{ar}_B], \\ \mathsf{ref}_{A\times B} &= [\mathsf{ref}_A, \mathsf{ref}_B], \end{aligned}$$

where for functions $f : P \longrightarrow R$ and $g : Q \longrightarrow R$, $[f, g] : P{+}Q \longrightarrow R$ is defined on $x \in P{+}Q$ as $f(x)$ or $g(x)$ according as $x \in P$ or $x \in Q$ respectively. Note that the range of a includes $\star$.

Function space. Morally $A \Rightarrow B$ is obtained by identifying the initial actions of B with the initialising action of A, in other words allowing B to use A as a resource:

$$\begin{aligned} \mathsf{Act}_{A\Rightarrow B} &= \mathsf{Act}_A + \mathsf{Act}_B, \\ a \vdash_{A\Rightarrow B} b &\iff (a \neq \star \text{ and } a \vdash_A b) \text{ or } a \vdash_B b \\ &\qquad \text{or } (\star \vdash_A b \text{ and } \star \vdash_B a), \\ \mathsf{ar}_{A\Rightarrow B} &= [\mathsf{ar}_A, \mathsf{ar}_B], \\ \mathsf{ref}_{A\Rightarrow B} &= [\mathsf{ref}_A, \mathsf{ref}_B], \end{aligned}$$

but since this labelled graph is not a tree, we make a separate copy of A underneath each of the initial actions of B. Thus we change $\mathsf{Act}_{A\Rightarrow B}$ to be $\mathsf{Act}_A \times I + \mathsf{Act}_B$, where I is the set of initial actions of B, and duplicate the enabling and labelling structure accordingly. See HO for a similar construction.

Universal quantification. Given $n \geq 0$ and $n{+}1 \vdash A$ the 1-arena $n \vdash \forall_n(A)$ is obtained by incrementing the arity of each initial vertex of A to "create a new hole", and then "binding" every occurrence of the reference $\star_{n+1}$. For $a \in \mathsf{Act}_A$ let $\tilde{a}$ be the unique initial action lying on the path from $\star$ to a, and define $m_a = \mathsf{ar}_A(\tilde{a}) + 1$.

$$\begin{aligned} \mathsf{Act}_{\forall_n(A)} &= \mathsf{Act}_A, \\ a \vdash_{\forall_n(A)} b &\iff a \vdash_A b, \\ \mathsf{ar}_{\forall_n(A)}(a) &= \begin{cases} \mathsf{ar}_A(a) + 1, & \text{when } \star \vdash_A a \\ \mathsf{ar}_A(a), & \text{otherwise} \end{cases} \\ \mathsf{ref}_{\forall_n(A)}(a) &= \begin{cases} (\tilde{a}, m_a), & \text{if } \mathsf{ref}_A(a) = \star_{n+1} \\ \mathsf{ref}_A(a), & \text{otherwise.} \end{cases} \end{aligned}$$

As an example, $\llbracket Bool \rrbracket = \forall_0(A)$ where A is the 1-arena:

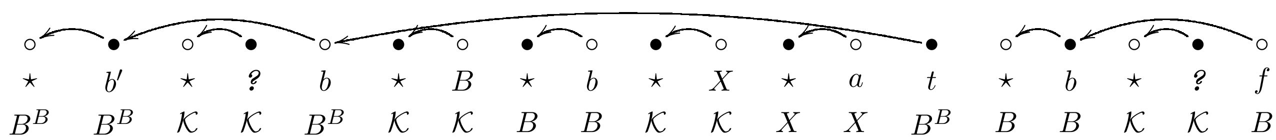
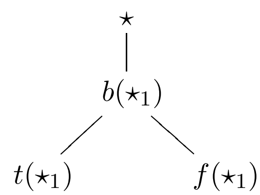


Figure 2. One of the four winning positions of $[\![\, not \,]\!]$ on $G([\![\, Bool \to Bool \,]\!], 0)$.



Equivalence. Write $A \sim B$ if there exists a bijection $\theta : \mathsf{Act}_A \to \mathsf{Act}_B$ that preserves and reflects structure, an equivalence relation respected by the previous three constructions.

Substitution. We define substitution of 1-arenas for references $\star_i$ of 1-arenas. It is easiest just to poach the definition of substitution from the type syntax:

Lemma 3 *Every system F type in context $[X_1, X_2, \ldots, X_n] \vdash T$ has an interpretation as a finite 1-arena in context $n \vdash [\![\, T \,]\!]$. Conversely any finite 1-arena in context $n \vdash A$ defines a type in context $[X_1, X_2, \ldots, X_n] \vdash \hat{A}$ such that $[\![\, \hat{A} \,]\!] \sim A$.*

Proof: The first half of the lemma is by structural induction on the type. $[\![\, unit \,]\!]$ is $\langle \emptyset, \emptyset, \emptyset, \emptyset \rangle$ and $[\![\, X_i \,]\!]$ is given by $\mathsf{Act}_{[\![\, X_i \,]\!]} = \{a\}$, $\star \vdash_{[\![\, X_i \,]\!]} a$, $\mathsf{ar}_{[\![\, X_i \,]\!]}(a) = 0$ and $\mathsf{ref}_{[\![\, X_i \,]\!]}(a) = \star_i$, for some $a \in \mathcal{U}$. Product, function space and Π-types are interpreted by the product, function space and universal quantification of 1-arenas respectively.

The converse is by recursion on the depth of the forest of A. For $\mathsf{Act}_A = \emptyset$ define $\hat{A} = unit$. For A of non-zero depth, first consider the case A has a unique initial token a. Let k be the arity of a and let $a^1, a^2, \ldots, a^m$ be the direct descendents of a. Define A^i to be the 1-arena with $\mathsf{Act}_{A^i} = \{\, b \mid a^i \vdash_A^* b \,\}$ and structure inherited from A, but with references a_j replaced by $\star_{n+j}$. Being of strictly smaller depth they define types $[X_1, \ldots, X_{n+k}] \vdash \hat{A^i}$. Now define

$$\hat{A} := \Pi X_{n+1} \ldots X_{n+k}.(\hat{A^1} \to \ldots \to \hat{A^m} \to X_l),$$

where $l = i$ when $\mathsf{ref}_A(a) = \star_i$ and $l = n + j$ when $\mathsf{ref}_A(a) = a_j$. Since in general any 1-arena A is the product of 1-arenas $B^1, \ldots, B^k$ each with a unique initial vertex (inspect the definition of product of 1-arenas), we define

$$\hat{A} = \hat{B}^1 \times (\ldots \times (\hat{B}^{k-2} \times (\hat{B}^{k-1} \times \hat{B}^k)) \ldots). \quad \square$$

Given $n \vdash A$ and an n-tuple $\theta = [A^1, \ldots, A^n]$ of 1-arenas $m \vdash A^i$, the ***substitution*** of A by θ, $A \circ \theta$, is the interpretation of the type $\hat{A}[\hat{A^1}/X_1, \ldots, \hat{A^n}/X_n]$. Thus $m \vdash A \circ \theta$.

7 The fibred category $\mathbb{F}$ of games

First define an auxiliary fibration $p : \mathbb{G} \to \mathbb{B}$ with $Ob(\mathbb{B}) = \mathbb{N}$ and $Ob(\mathbb{G}_n)$ the set of finite 1-arenas in context n. A morphism in $\mathbb{B}(m, n)$ is an n-tuple of objects of $\mathbb{G}_m$. Composition[3] is substitution, product is addition, and the terminal object is 0. Define a ***starting strategy*** on $G(A, n)$ to be a strategy that contains $start(A, n)$. A morphism in $\mathbb{G}_n(A, B)$ is a starting strategy on $G^{fin}(A \Rightarrow B, n)$, the subgame of $G(A \Rightarrow B, n)$ obtained by restricting all 2-actions to be finite 1-arenas.

Composition of strategies is a "second-order" version of HO-projection from a history of interaction. With condition $F4$ and the related concepts outlined in the introduction of section 5 the rest is just a combinatorial grind, and there is only one way to proceed.

The fibre $\mathbb{G}_n$ is a CCC with product and function space as given in the previous section. Identities are copy-cat strategies as usual, extended to second order by the fact the i^{th} occurrence of the 2-action *?* made by O is copied by P as the singleton arena $a(\star_i)$. The generic object is $a(\star_1) \in \mathbb{G}_1$.

[3] As stated here in the informal outline of the fibration, composition is only associative up to $\sim$. Further technical details are to be found in the author's thesis [Hug].

Cartesian maps are based on a form of copy-cat strategy. Indexed products are provided by universal quantification of 1-arenas, and the verification of the adjunction is trivial because the first $2n+3$ moves of any sequence on $\forall_n(A)$ are a simple permutation of $\mathit{start}(A, n+1)$.

This structure provides an interpretation of system F as detailed in [Pho]. We now present a lemma which will help us to construct our model $\mathbb{F}$. The proof is by structural induction on normal forms.

Lemma 4 *Every normal form is interpreted in $\mathbb{G}$ as a winning strategy.*

The proof of the theorem below is similar in flavour to the HO definability proof.

Theorem *Given any finite 1-arena in context $n \vdash A$, every winning strategy σ on $G^{fin}(A, n)$ defines a system F η-expanded, β-normal form s_σ of type $\hat{A}$, and the interpretation of s_σ in $\mathbb{G}$ is σ.*

Proof: By recursion on the size of σ (our games are finitely branching at O-moves, so winning strategies are finite, by König's Lemma). If $\sigma = \mathit{start}(A, n)$ then A is necessarily the empty 1-arena, and we define $s_\sigma = \langle\rangle$ of type $\hat{A} = \mathit{unit}$. First consider the case A has a unique initial action a, with direct descendents $b_{11}b_{12}, \ldots, b_{1p_1}$. Let $\vec{\alpha} = \star ? \star ? \ldots \star ?$ of length $2m$, where m is the arity of a. The initial segment of play is of the form $\mathit{start}(A, n)\, a\, \vec{\alpha}\, \vec{\beta_1} \star \vec{\beta_2} \star \ldots \star \vec{\beta_d}$ where $\vec{\beta_i} = b_{iq_1} \star A_{i1} \star A_{i2} \ldots \star A_{in_1}$ and, for $2 \le i \le d$, $b_{i1}, b_{i2}, \ldots, b_{ip_i}$ are the initial actions of the 1-arena $A_{(i-1)e_i}$ for some $1 \le e_i \le n_{i-1}$.

We construct an arena A' such that there is a 1-1 correspondence between possible continuations of the rest of the game on A above, and complete games on A'. Let $a_{i1}, \ldots, a_{ir_i}$ be the direct descendents of b_{iq_i}. Then define

$$A' = \prod_{i,j} (C \Rightarrow F_{ij}),$$

where the product symbol indexed by $1 \le i \le d$ and $1 \le j \le r_i$ is the product of 1-arenas, and $C = C_1 \times C_2 \times \ldots \times C_{p_1}$. C represents the continuations that are available to P in the future by justifying back to O's initial move a, and F_{ij} represents the possible continuations of A that would result from O playing the token a_{ij} directly descendent from b_{iq_i}. The construction of C and of the F_{ij} involves the substitution of 1-arenas for variables, as defined in section 6 (details omitted).

By the 1-1 correspondence stated above, σ on the rest of A determines a winning strategy σ' on A', which is equivalent to a strategy σ'_{ij} on each of the 1-arenas $(C \Rightarrow F_{ij})$. Since σ' is strictly smaller than σ, by the induction hypothesis we have normal forms s_{ij} defined by the σ'_{ij}.

We now define the term s_σ. For $2 \le i \le d$ define the context

$$\pi_i[\ldots] = \begin{cases} \mathit{Snd} \ldots \mathit{Snd}\, \mathit{Fst}[\ldots], & \text{if } 1 \le q_i < p_i, \\ \mathit{Snd} \ldots \mathit{Snd}\, \mathit{Snd}[\ldots], & \text{if } q_i = p_i, \end{cases}$$

where the number of consequtive "Snd"s in the first case is $q_i - 1$, and in the second case is q_i, and the context

$$E_i[\ldots] = (\pi_i[\ldots])\, \hat{A}_{i1} \hat{A}_{i2} \ldots \hat{A}_{in_i} s_{i1} s_{i2} \ldots s_{ir_i}$$

The normal form s_σ is given by

$$\Lambda X_1 X_2 \ldots X_m . \lambda b_{11}{}^{\hat{C}_1} . \lambda b_{12}{}^{\hat{C}_2} \ldots \lambda b_{1p_1}{}^{\hat{C}_{p_1}} . v$$

where

$$v = E_d E_{d-1} \ldots E_2 [a_{1q_1} \hat{A}_{11} \ldots \hat{A}_{1n_1} s_{11} s_{12} \ldots s_{1e_1}].$$

The verification that the interpretation of s_σ is σ comes from within the proof of Lemma 2.

Finally, in the case that A has more than one initial action, we take the terms that are generated by restriction of σ to each of the components, and then form the appropriate $\langle\ ,\ \rangle$-pairings of these terms that correspond to the product type $\hat{A}$ as defined in Lemma 3. $\square$

Now we define $\mathbb{F}$ as the subcategory of $\mathbb{G}$ obtained by reducing the hom-sets in the fibres to the winning strategies. To show that the composition of winning strategies is winning, we lift them to System F (Theorem), sequentially compose the terms, normalise, and reinterpret back into the model (Lemma 4).

8 Conclusions and future research

Interestingly, the definability theorem does not provide a one-to-one correspondance between winning strategies and η-expanded, β-normal forms, since the interpretation map induces the following equations on types (and terms include types in general):

$$\begin{array}{rclrcl} unit \times T & = & T, & unit \to T & = & T, \\ T \to unit & = & unit, & \Pi X.unit & = & unit, \end{array}$$

$$\begin{array}{rcl} T_1 \to T_2 \to T_3 & = & T_1 \times T_2 \to T_3, \\ T_1 \to T_2 \times T_3 & = & (T_1 \to T_2) \times (T_1 \to T_3), \\ \Pi X.\, T_1 \times T_2 & = & (\Pi X.T_1) \times (\Pi X.T_2), \\ T_1 \to \Pi X.T_2 & = & \Pi X.\ T_1 \to T_2, \end{array}$$

where in the last axiom X is not free in T_1.

Work in progress that has stemmed from this model:

- Adaptation of the games in an attempt to obtain a Reynold's parametric model of system F [Rey83].
- In order to model system F using our hierarchy we required only the use of one particular 2-arena, the "flat" arena $\mathcal{K}$. Can we obtain applicative structure at second order by allowing arbitrary 2-arenas, so that we can model higher-order polymorphism [Cro95]?
- Can we adapt the games in order to model linear polymorphism [Abr95]?

Future work:

- Give a direct proof of compositionality of winning strategies.
- Can we derive the copy-cat condition from a more general notion of game, rather than enforcing it?
- Investigate the abstract machine that corresponds to composition in the model.
- Can we model dependent types with a dual version of 2-reference which relates 2-arenas back to previous 1-arenas?

A long-term objective is to try and adapt the modelling techniques used in this paper in order to conquer every vertex of the λ-cube [Bar84].

Acknowledgements Many thanks to Luke Ong, who originally suggested polymorphism as a thesis topic in March 1996. Also to Martin Hyland, Ralph Loader and Guy McCusker for helpful discussions and comments. Finally Ben Worrell and Adèle Carney for their help with LaTeX.

References

[Abr95] S. Abramsky. Semantics of Interaction. Notes for CLICS II Summer School, Cambridge. Summer 1995.

[AJM94] S. Abramsky, R. Jagadeesan, and P. Malacaria. Full abstraction for *PCF* (extended abstract). Number 789 in Lecture Notes in Computer Science, 1994.

[Bar84] Henk P. Barendregt. *The Lambda Calculus: Its Syntax and Semantics*, North-Holland, 1984.

[CGW89] T. Coquand, C. Gunter and G. Winskel. Domain theoretic models of polymorphism. *Information and Computation,* 81:123-167, 1989.

[Cro93] R. L. Crole. *Categories for types.* Cambridge University Press, 1993.

[Gir89] J.-Y. Girard. *Proofs and Types.* Cambridge Tracts in Theoretical Computer Science, 45:159-192, 1989.

[HO94] J. M. E. Hyland and C.-H. L. Ong. On Full Abstraction for PCF: I, II and III. ftp-able at ftp.comlab.ox.ac.uk, 1994.

[Hug] D. J. D. Hughes. Games, polymorphism and parametricity. DPhil thesis, University of Oxford.

[McC96] G. McCusker. Games and Full Abstraction for FPC. *LiCS* 1996.

[Nic94] H. Nickau. Hereditarily sequential functionals. Proceedings of the Symposium on Logical Foundations of Computer Science: Logic at St. Petersburg. Springer, 1994.

[Pho] W. Phoa. An introduction to fibrations, topos theory, the effective topos and modest sets. Lecture Notes, University of Edinburgh, Scotland, U.K.

[Rey83] J. C. Reynolds. Types, abstraction, and parametric polymorphism. Information Processing 83:513-523. North-Holland, 1983.